**MICHEL ARCHAMBAULT**

# PROGRAMMES UTILITAIRES POUR AMSTRAD




SORACOM informatique

# PROGRAMMES UTILITAIRES POUR AMSTRAD

MICHEL ARCHAMBAULT

# PROGRAMMES UTILITAIRES POUR AMSTRAD

SORACOM informatique

*Du même auteur :*

*à ETSF :*
*— Labo Photo - Montages électroniques*
*— Guide pratique des montages électroniques*
*— Construisez et perfectionnez vos appareils de mesures*
*— Formation pratique à l'électronique moderne*

*à la SORACOM :*
*— Mieux programmer sur ORIC-1 et ATMOS*
*— Mieux programmer sur AMSTRAD*

# TABLE DES MATIERES

# INDEX PROGRAMMES/CHAPITRES

# COURTE PRÉFACE

Votre AMSTRAD sait aussi faire autre chose que de ''lasériser'' des extra-terrestres ; il sait se rendre utile pour résoudre vos petits problèmes, ou écourter de façon spectactulaire des besognes courantes et fastidieuses. Reste la question du logiciel : ceux du commerce sont souvent très difficiles à adapter à votre besoin précis, voire inexistants. Il faut le programmer soi-même.

Ce livre veut être un utilitaire pour faire des utilitaires ; il contient davantage de méthodes, d'astuces, de mises en garde, que de logiciels genre ''PRET-A-LOADER''. En revanche, beaucoup de routines courtes et passe-partout d'usage fréquent et des programmes plus longs mais conçus pour être modifiés en fonction des besoins.
A l'exception d'un seul, tous nos programmes sont en Basic, donc à la portée de tous.

# Chapitre I

# DES PRÉLIMINAIRES IMPORTANTS

Afin que vous puissiez bien tirer parti de ce livre, il est indispensable que nous définissions ce que seront la forme et le fond ; non seulement sur le plan général mais aussi concernant des points de détail très précis.

## A QUI, CE LIVRE S'ADRESSE-T-IL ?

A ceux que l'on appelle des amateurs, c'est-à-dire des personnes considérant la micro-informatique comme un jeu intellectuel ; ce qui n'exclut pas des prolongements vers la vie professionnelle. Très nombreux sont les travailleurs indépendants qui, après avoir eu le virus de la micro, se sont informatisés tout seuls avec leurs propres logiciels. Nombreux aussi ceux qui ont quitté leur ancien métier pour devenir informaticien à part entière. Ce dernier cas est celui de l'auteur. Voilà qui devrait vous rassurer : être clair et efficace sont les deux impératifs.

Les acheteurs de micro-ordinateurs se divisent en gros en trois catégories : ceux que seuls les jeux du commerce intéressent. Ceux qui ont besoin d'un micro dans un but professionnel et qui utilisent pour cela des logiciels ''tous faits'' (traitement de texte, comptabilité, etc.). Enfin, les créatifs, les chercheurs, ceux qui aiment construire un programme qui matérialise une idée, c'est-à-dire vous (sinon vous n'auriez pas acheté ce livre... ).
L'auteur sait donc qu'il s'adresse à des ''mordus'', mais que certains d'entre vous sont encore un peu débutants, tandis que d'autres sont incollables sur les fonctions Basic de l'AMSTRAD. Comment concilier des niveaux si différents ? Les ''forts'' pardonneront quelques rappels de choses qu'ils savaient déjà, alors que pour les ''faibles'', nous ferons parfois référence à un ouvrage du même auteur ''MIEUX PROGRAMMER SUR AMSTRAD'' (Editions SORACOM), qui est, en quelque sorte, la *suite pratique* du manuel d'origine. Ce sera de la forme ''(voir MPSA p 117)''.

## QUELS UTILITAIRES ?

Ce mot désigne en fait trois choses différentes :

1) La petite routine qui trouve donc sa place dans trois programmes sur cinq. C'est généralement un sous-programme appelé par GOSUB. C'est une routine dite standard.
2) Le programme de un ou dix kilo-octets, bien indépendant et faisant une tâche bien précise. Il est généralement ''truffé'' de routines standard.
3) Le même que précédemment, mais en logiciels du commerce : il y a ceux qu'il vaut mieux acheter, comme les traitements de texte (AMLETTRE, TEXTOMAT, etc.) ou les tableurs (MASTER CALC, MULTIPLAN, etc.), et ceux qu'il vaut mieux faire soi-même, parce que le ''sur-mesures'' va ridiculiser en performances le logiciel ''en confection''. Le cas typique est celui des gestions de fichiers.

Cela tombe bien puisqu'un sondage récent a montré que la grande majorité des utilisateurs d'AMSTRAD était surtout intéressée par la gestion de fichiers !

## ALORS, QU'ALLONS-NOUS TROUVER DANS CE LIVRE ?

Beaucoup de routines, peu de programmes complets (ils seront relativement courts), mais en revanche beaucoup de ''marches à suivre'' avec exemples, pour que vous construisiez **votre** programme à l'aide de ces routines ; routines que vous devrez parfois modifier (elles sont expliquées).
En effet, il serait malhonnête de la part de l'auteur de publier un listing de vingt pages qui aurait peu de chances de vous convenir parfaitement. Mieux vaut vous livrer les pièces détachées et **vous apprendre à les assembler,** dans le but d'obtenir un logiciel correspondant à vos besoins personnels et précis. En informatique, comme dans le bâtiment, c'est souvent plus rapide de faire du neuf que de retaper du vieux.
Tous les listings présentés ont été *sélectionnés* parmi ceux que l'auteur réutilise plusieurs fois par an. Donc, il est à peu près certain que vous aurez l'usage de chacun d'entre eux.
Pour plus de clarté, nous avons réparti nos chapitres en quatre domaines :

1) Les utilitaires de programmation. Donc tout ce qui touche à l'écriture en général.
2) Les utilitaires graphiques : tracés de courbes, diagrammes, axes, etc.
3) La gestion de fichiers. Peu comparable avec les logiciels du commerce ; plus courts et bien plus performants. Nous verrons pourquoi.
4) Les utilitaires pour imprimante. Copie d'écran, édition de tableaux, d'étiquettes, etc.

Certes, ce n'est pas complet. Il n'y a pas de programmes utilitaires pour la musique et pas grand chose en langage machine. Il fallait faire des choix en fonction des demandes prioritaires.
Et le style ? Essayons d'être clairs pour être compris, donc sans entrer dans des détails peu utiles à connaître. Ce n'est surtout pas un ''livre d'école'', on est en loisir-détente.

## QUELQUES PETITES NORMES

### Les listings

Ils sont imprimés en 40 caractères par ligne, exactement comme à l'écran après avoir fait LIST. Le but est de vous fournir des repères afin de dénicher une faute de frappe.
En général, nos programmes commencent en ligne 10, mais certains entre 50000 et 65000 : ce sont des routines utilitaires terminées toujours par un RETURN. Elles seront sauvegardées sur une disquette ou cassette à part, nommée ''UTIL'' d'où elles seront rappelées par MERGE pour être logées dans le programme en cours d'écriture.
ATTENTION ! pour les CPC 464, ces routines Basic seront enregistrées en ASCII. Exemple :

SAVE''MENU'',A

sinon le risque est grand d'avoir un ''EOF met'' au cours du MERGE''MENU'' ; et perte du programme en RAM ! Ce bug du 464 a été supprimé dans les 664 et 6128.

### Les erreurs

Les programmes publiés sont comme on dit ''éprouvés'' et ont été listés sur l'imprimante de l'auteur (EPSON RX80). Seuls risques : des manques ou taches d'encrage lors de l'impression du livre.
Le courrier des lecteurs de la revue ''CPC'' nous a montré les classiques fautes de frappe qui font dire ''le programme ne fonctionne pas'' :

- Confusion des lettres I, l minuscule, chiffre 1 et barre verticale (SHIFT + a commercial).
- Confusion du ''O'' et du zéro.
- Confusion de ponctuations : point-virgule et deux points ou point et virgule.
- Oubli ou ajout d'un blanc (espace). Le plus classique étant le symbole de ''Chaîne vide'' : deux guillemets consécutifs et non pas guillemet, blanc, guillemet...
- Ecriture du programme en fonction AUTO, sans remarquer que l'auteur fait souvent des sauts de numérotation à des mutliples de 100 ou de 1000, pour la clarté du listing ; et en recopiant scrupuleusement le contenu des lignes, comme un ''GOTO 350'', ce qui peut alors avoir des conséquences imprévues...

Ce recueil d'étourderies constitue les 90 % des ''programmes qui ne marchent pas''. (Pour ''débuguer'' un programme, voir MPSA p 101 à 106.)

**Quels micros AMSTRAD ?**

L'auteur utilisant un CPC 464 couleur équipé d'un lecteur de disquettes DDI-1, il n'y a aucun problème de compatibilité avec les modèles 664 et 6128. D'autre part, nous avons vérifié la compatibilité avec le CPC 464 possédant uniquement le lecteur de cassettes (sauf exceptions signalées à l'avance).
En revanche, nous serons moins sûrs de nous en ce qui concerne les écrans monochromes : en retapant les listings, méfiez-vous des PAPER et PEN 2 ou 3 et des changements d'INK. Mais c'est là un genre de problème dont vous devez déjà être coutumier.

**Quelle imprimante ?**
Il y a, disons plutôt il y a eu, la DMP-1 et les ''autres''. Ces autres ont des codes de commandes conformes au ''standard EPSON'', à de très rares exceptions près. La DMP-1 (ou son homologue SEIKOSHA) a très peu de codes et ils ne sont pas standard. Si, dans un listing, il y a PRINT #8,A$ pas de problème avec la DMP-1, mais à partir du moment où l'on sort de l'ordinaire avec des CHR$(27), cela ne marchera qu'avec les modèles standard (DMP 2000, STAR, EPSON, OKI, CENTRONIC, etc.).
Si vous possédez une imprimante non-standard, vous ne serez pas lésé, car il vous suffira de traduire nos codes. Pour cela, la routine 59000 du chapitre II sera pleine d'enseignements pour les commandes courantes. Pour les codes plus spéciaux, nous indiquerons toujours leurs effets, afin que vous puissiez trouver l'équivalent dans le manuel de votre imprimante.

Ce qui fait la force des AMSTRAD CPC, outre le rapport performances/prix, est le respect généralisé des divers standards : le Basic, le DOS CPM, le clavier, le magnétophone et l'écran de 25 lignes en 40 ou 80 caractères. Il doit en être de même pour l'imprimante. En espérant que la disquette trois pouces devienne un standard car elle *est bien plus pratique* que les 5 pouces 1/4 ou les 3 pouces 1/2.

**Nos programmes ''complets''**

Ils sont destinés à vos coups de ciseaux, soit pour en prélever la partie qui vous intéresse en y modifiant quelques lignes, soit, au contraire, pour les grossir. Si le but de l'auteur avait été de livrer des gros programmes utilitaires d'au moins 10 Ko chacun, il n'aurait pas choisi le livre mais la cassette ou disquette.
Si ce n'est déjà fait, il est absolument obligatoire de vous familiariser avec les commandes Basic RENUM, DELETE et MERGE.
Pour votre programme final, évitez le genre ''fleuve'' (bout à bout), mais faites de la programmation ''structurée'' à base de GOSUB (voir MPSA chapitre III).
Une fois cette habitude de découpages et recollages à partir des ''archives'' prise, vous ne serez plus surpris de faire en une seule soirée un 6 kilo-octets qui tienne la route.

# Chapitre II

# L'ÉCRITURE RAPIDE

Le but n'est pas d'acquérir une virtuosité des doigts sur le clavier, mais d'éviter des gestes inutiles ou de ré-improviser toujours les mêmes lignes Basic. On gagne alors en rapidité et en fiabilité.

## LE PROGRAMME KEY

C'est une version améliorée de celui publié dans ''Mieux programmer sur AMSTRAD''. Ce court programme est à lancer par RUN à chaque fois que l'on s'apprête à programmer ou à retoucher un programme. C'est une bonne habitude à prendre car le gain de temps est considérable, et ce pour deux raisons :

— Il n'est plus nécessaire d'appuyer sur SHIFT pour obtenir des caractères aussi fréquents que $, =, '' '', (ou). On peut donc taper avec la main droite tandis que l'index de la main gauche suit le listing à copier.
— Le pavé des touches-clés est programmé pour écrire des longueurs Basic fréquentes, telles que LOCATE, CHR$(, etc. Certaines d'entre elles comprennent l'exécution (+CHR$(13)=touche ENTER).

La ligne 210 est une suite de sages précautions ; elle réinitialise aux valeurs par défaut les paramètres d'affichage. En effet, on pourrait hériter d'autres valeurs provenant d'un programme précédent ; elles ne seraient pas annulées par NEW, CLEAR ou RUN.

Le listing n'utilisant pas d'astuces particulières, abordons tout de suite le mode d'emploi.
A l'écran apparaît la signification des touches du pavé numérique, à l'exception de la petite touche ENTER (trop utile sur le CPC 464 et absente sur le CPC 6128).
Au-dessous de cet encadré, les numéros de codes de ces touches. Ils seront utiles si l'on désire redéfinir une de ces touches. A noter que la touche ''6'' est laissée libre. Vous voulez lui faire écrire ''DATA''. Frappez cette touche, il apparaît :

KEY 134

Il vous suffit alors de compléter par :

,"DATA " et ENTER

En bas de l'encadré, la liste des caractères redéfinis :
La touche TAB donne le signe =.
Les crochets deviennent parenthèses.
Le a commercial fournit le guillemet.
L'anti-slash (à droite du ?/) = $.
Tapez alors la lettre N (ou n), cela effectue un CLS et un NEW, mais tout reste défini !

*QUELQUES REMARQUES*

- Le signe "flèche en haut" apparaissant dans l'encadré signifie que le CHR$(13) est inclus, exemple, la touche "7" efface l'écran, la touche "4" conduit aux lettres noires sur fond bleu-ciel (pour le MODE 2), donc une touche au lieu de 17 à frapper...
- La plupart des fonctions Basic du pavé numérique sont suivies d'un blanc, c'est le cas pour LOCATE, EDIT, LIST, PRINT et FOR I = 1 TO. Cela évitera bien des "Syntax Error".
- La disposition de ces touches de fonction n'est pas due au hasard. Vous apprécierez à l'usage le fait que LIST soit juste au-dessus de la petite touche ENTER.

En plus de votre disquette utilitaire, n'oubliez pas de toujours recopier le programme KEY sur votre disquette de travail.

| | | |
|---|---|---|
| 135 | 136 | 137 |
| 132 | 133 | 134 |
| 129 | 130 | 131 |
| 128 | 138 | |

| | | |
|---|---|---|
| CLS↑ | FOR I=1 TO | EDIT |
| Noir/Ciel↑ | Jaune/Bleu↑ | libre |
| LOCATE | CHR$( | LIST |
| PRINT | SAVE " | |

TAB est le signe =
Crochets = Parentheses
A commercial = Guillemets
anti-slash = $

TAPEZ N POUR NEW + CLS

```
10 ' *** KEY ***
20 'AMSTRAD CPC -  M.Achambault - 1986
30 '  -- Non modifiees par NEW --
40  KEY 128,"PRINT ":' 0
50  KEY 129,"LOCATE ":' 1
60  KEY 130,"CHR$(":' 2
```

```
70  KEY 131,"LIST ":' 3
80  KEY 132,"INK 0,20:INK 1,0"+CHR$(13):
' 4
90 KEY 133,"INK 0,1:INK 1,24:BORDER 1"+C
HR$(13):' 5
100 KEY 134, "KEY 134":' 6
110 KEY 135,"CLS"+CHR$(13):' 7
120 KEY 136,"FOR I=1 TO ":' 8
130 KEY 137,"EDIT ":' 9
140 KEY 138,"SAVE "+CHR$(34)
150 KEY DEF 17,0,40:' (
160 KEY DEF 19,0,41:' )
170 KEY DEF 22,0,36:' $
180 KEY DEF 26,0,34:' "
190 KEY DEF 68,0,61:' =
200 'AFFICHAGE
210 MODE 1:WINDOW #0,1,40,1,25:CLS:ORIGI
N 0,0:ZONE 13:PAPER 0:PEN 2
220 FOR I=135 TO 129 STEP -3:PRINT TAB(5
);I;TAB(19);I+1;TAB(32);I+2:NEXT
230 PRINT TAB(5);128;TAB(19);138:PEN 1
240 LOCATE 5,7:PRINT "CLS^";:PRINT TAB(1
6);"FOR I=1 TO";:PRINT TAB(33);"EDIT"
250 LOCATE 3,10:PRINT "Noir/Ciel^";:PRIN
T TAB(16);"Jaune/Bleu^";:PRINT TAB(32)"l
ibre"
260 LOCATE 4,13: PRINT "LOCATE";:PRINT T
AB(19);"CHR$(";:PRINT TAB(33);"LIST"
270 LOCATE 4,16: PRINT "PRINT";SPC(10);"
SAVE "+CHR$(34)
280 LOCATE 12,19: PRINT "TAB est le sign
e ="
290 LOCATE 10,20:PRINT "Crochets = Paren
theses"
300 LOCATE 8,21: PRINT  "A commercial =
Guillemets"
310 LOCATE 14,22:PRINT "anti-slash = $"
320 LOCATE 10,24:PRINT "TAPEZ N POUR NEW
 + CLS"
330 PLOT 5,324:DRAWR 630,0,2
340 DRAWR 0,-190
350 DRAWR -630,0:DRAWR 0,190
360 WHILE R$="":R$=INKEY$:WEND
370 IF UPPER$(R$)="N" THEN CLS:NEW
380 END
390 '-------- FIN DE LISTING ---------
```

## LE PROGRAMME DISCUT

Ce petit utilitaire pour DISC va ''faire le ménage'' sur vos faces de disquettes encombrées, et sans erreurs de syntaxe possibles.
Toutes les disquettes de l'auteur, comportant des programmes Basic ou des fichiers, ont une copie de DISCUT.
Une fois lancé, CLS, et en haut de l'écran apparaît cette ligne menu :

Cat Erase Rename Drive Bak- I Quit

Il vous suffit de frapper la lettre initiale (majuscule ou minuscule), sans ENTER, pour que l'ordre s'exécute.
CAT : c'est banal.
ERASE : On vous demande le nom *complet* du programme ou du fichier à effacer. Si vous tapez *.* (=tous !), l'écran vous demande confirmation.
RENAME : L'écran vous demande l'ancien nom, ENTER, puis le nouveau nom.
DRIVE : Pour ceux d'entre vous possédant deux drives. L'écran demande sur quel drive, A ou B, vous voulez travailler.
BAK- : Très utile ! Il suffit donc de presser la touche B pour éraser tous les fichiers et programmes. BAK.
Après exécution d'un de ces ordres, la ligne menu est de nouveau proposée (sans CLS).
I (La barre par SHIFT/a commercial) : Vous fait ICPM, donc avec perte du programme Basic.
QUITTER : C'est un END, le programme reste en RAM.

Le module commençant en ligne 1000 va vérifier si vous n'avez pas fait de fautes de frappe en entrant un nom de fichier. Si c'est le cas, BIP sonore, et vous revenez à la ligne de menu.
Du super luxe !

```
10 '*** DISCUT *** Utilitaire pour DISC
20 'AMSTRAD CPC - M. Archambault
50 CLS
100 PRINT "Cat  Erase  Rename  Drive  Ba
k-  !  Quit"
110 Q$="":WHILE Q$="":Q$=UPPER$(INKEY$):
WEND
120 Q=INSTR("CERDB!Q",Q$):IF Q=0 THEN PR
INT CHR$(7);:GOTO 110
130 ON Q GOSUB 200,300,400,500,600,700,8
00
140 GOTO 100
```

```
200 CAT:RETURN
300 INPUT "  ERASER quel Fichier ? ",F$:
GOSUB 1000:IF REFUS THEN 390
320 IF F$="*.*" THEN PRINT"  Etes-vous b
ien sur ?...(O/N) ";:R$="":WHILE R$="":R
$=UPPER$(INKEY$):WEND:IF R$="N" THEN 390
 ELSE IF R$<>"O" THEN 320
330 |ERA,@F$
390 RETURN
400 INPUT "  ANCIEN NOM : ",F$:IF INSTR(
F$,"*") >0 THEN 400
410 GOSUB 1000:IF REFUS THEN 490 ELSE A$
=F$
420 INPUT " NOUVEAU NOM : ",F$:IF INSTR(
F$,"*") >0 THEN 420
430 GOSUB 1000:IF REFUS THEN 490 ELSE N$
=F$
440 |REN,@N$,@A$
490 RETURN
500 PRINT "  SUR QUEL DRIVE (A/B) ?"
510 D$="":WHILE D$="":D$=UPPER$(INKEY$):
WEND
520 IF D$<>"A" AND D$<>"B" THEN 510
530 |DRIVE,@D$
590 RETURN
600 F$="*.BAK"
610 |ERA,@F$
620 RETURN
700 |CPM
710 RETURN
800 END
1000 'SECURITE DU NOM DE FICHIER
1010 L=LEN(F$):REFUS=0
1020 IF INSTR(F$,CHR$(32)) >0 OR L=0 THE
N REFUS=1
1030 P=INSTR(F$,".")
1040 IF P>9 OR (P=0 AND L>8) THEN REFUS=
1
1050 IF P>0 AND L-P>3 THEN REFUS=1
1090 IF REFUS THEN PRINT CHR$(7);
1100 RETURN
1200 ' -------- FIN DE LISTING --------
```

## DEUX ROUTINES TRES COURANTES (MENU ET AZERTY)

Il s'agit de deux sous-programmes très courts mais d'un usage très fréquent, surtout le premier. Après les avoir tapés, sauvegardez-les séparément. Ils sont destinés à être rappelés par MERGE en cours d'écriture du programme. Cela explique leurs numéros de ligne très élevés. Rappel aux possesseurs de CPC 464 : Vous devez enregistrer ces programmes en ASCII, exemple SAVE''MENU'',A.
Pour le loger dans un programme en cours d'écriture, entrez en mode direct :

MERGE ''MENU''

## LA ROUTINE MENU

Vous avez souvent des questions posées à l'écran, un choix d'options A, B, C, X, Q, par exemple ; mais le programme ne doit pas planter si l'étourdi répond par ''Z'' ou par ''a'', d'autre part, il est déshonorant d'avoir à presser ENTER lorsqu'une réponse ne comporte qu'un seul caractère. Toutes ces sécurités sont en dix lignes Basic, toujours rappelées par un GOSUB 50000.
Le luxe va même jusqu'à afficher, en ligne 24, la liste des réponses possibles, et autocentrée (en MODE 1).
Mode d'emploi : Reprenons l'exemple des options A, B, C, X et Q. Programmez :

TEX$ = ''ABCXQ'':GOSUB 50000

En bas de l'écran apparaît :

Réponse (A, B, C, X, Q)→

Toute mauvaise réponse est signalée par un bip sonore. Si la réponse est valable, le sous-programme retourne un nombre K, représentant la place du caractère tapé dans la chaîne TEX$. Ainsi, pour X, on a K = 4. Vous pouvez poursuivre votre programme par :

ON K GOSUB 5000,6000,7000,8000,9000

ou toute autre exploitation de K.

```
50000 'REPONSE A UN MENU
50010 LT=LEN(TEX$)
50020 LOCATE 15-LT,24:PRINT"Reponse (";
50030 FOR I=1 TO LT-1
50040 PRINT MID$(TEX$,I,1);",";:NEXT
50050 PRINT RIGHT$(TEX$,1);")";CHR$(154)
;CHR$(243);CHR$(207)
```

```
50060 TEX$=UPPER$(TEX$)
50070 R$="":WHILE R$="":R$=INKEY$:WEND
50080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
50090 IF K=0 THEN PRINT CHR$(7);:GOTO 50
070
50100 RETURN
```

**LA ROUTINE AZERTY**

Ce GOSUB 51000 va permettre d'afficher les minuscules accentuées ainsi que les symboles ''numéro'' et ''paragraphe''. On obtient ces sept caractères français par les touches crochets, a commercial (avec ou sans SHIFT) et anti-slash.
Nous avons scrupuleusement respecté la norme ASCII internationale (pas comme dans les IBM PC...). De ce fait, ces lettres seront fidèlement reproduites sur imprimante, à condition que celle-ci ait reçu la commande ''clavier français'', à savoir :

PRINT # 8,CHR$(27);''R'',CHR$(1)

Les minuscules accentuées donnent un ''fini'' aux pages d'écran, mais cela présente souvent des inconvénients :

— Ce GOSUB doit être au tout début du programme, *avant* un OPENOUT''BIDON''.
— Le SYMBOL AFTER 64 de la ligne 51010 est encombrant en mémoire ; il peut y avoir antagonisme avec le chargement d'une routine en langage machine (rare mais authentique).
— Si vous lancez cette routine (pour l'essayer ou pour tester votre programme en cours d'écriture), alors que les touches définies par le programme KEY sont toujours actives, sachez que ces dernières (définies dans KEY par KEY DEF) sont prioritaires sur des définitions par SYMBOL. Ainsi, la touche anti-slash donnera toujours le signe $ et non le ''ç''.
  Donc, pour tester cette routine écrite avec KEY, faites SAVE''AZERTY'', initialisez la machine (CTRL + SHIFT + ESC), puis LOAD''AZERTY'' et enfin RUN (ou GOSUB 51000).
— Si vous listez un programme avec des chaînes comportant ces lettres, alors que la machine ou l'imprimante ne sont pas en ''caractères français'', il est normal que les ''é'' et les ''è'' soient remplacés par des petites accollades...
— Si ces SYMBOL sont actifs et que vous voulez taper ICPM, ou la barre pour appeler une routine en RSX, vous obtiendrez un ''ù'' à l'écran à la place de la barre. Mais, n'ayez aucune crainte, ça marche quand même ! Car l'ordinateur a seulement pris le code ASCII (124) de cette touche.

```
51000 ' AZERTY ACCENTUE
51010 SYMBOL AFTER 64
51020 SYMBOL 64,96,48,120,12,124,204,118
,0
51030 SYMBOL 91,0,56,108,56,0,0,0,0
51040 SYMBOL 92,0,0,60,102,96,102,60,24
51050 SYMBOL 93,60,96,60,102,60,6,60,0
51060 SYMBOL 123,12,24,60,102,126,96,60,
0
51070 SYMBOL 124,48,24,102,102,102,102,6
2,0
51080 SYMBOL 125,48,24,60,102,126,96,60,
0
51100 RETURN
```

## LES CODES D'IMPRIMANTES

C'est spectaculaire tout ce que l'on peut faire comme variétés de caractères d'impression ; quatre tailles de lettres, écriture fine, grasse, normale, italiques, etc. Mais, quand il s'agit de transcrire ces fastidieuses suites des CHR$(27), CHR$ de quelque chose, là, l'enthousiasme tombe... Surtout quand il faut compulser le manuel de l'imprimante plusieurs fois pour une même ligne Basic, tant ces codes sont impossibles à retenir.
La solution était sans doute trop simple, rebaptisons ces codes (en RAM) avec des noms évocateurs ; par exemple, l'ordre pour souligner les mots, c'est SOUL$ et celui pour l'annuler, c'est SOULX$. Le ''X'' final symbolisant une rature en croix.
Auparavant, on a défini :

SOUL$ = CHR$(27) + '' – '' + CHR$(1)

Ensuite, à chaque fois que l'on veut souligner, par exemple, le mot AMSTRAD :

PRINT #8,''Micro-ordinateur'';SOUL$;'' AMSTRAD'';SOULX$;'' CPC''

Tous les ordres usuels en codes EPSON sont écrits une fois pour toutes dans une routine ''EPSON'' que l'on joint à notre programme par MERGE ''EPSON''. Un GOSUB 59000 vers le début du programme, et voilà en mémoire RAM seize codes faciles à retenir et aussi très courts à écrire. Jugez plutôt :

EFF$ = Effacement du buffer de l'imprimante (initialisation).
GRAS$ = Ecriture grasse (renforcée).
GROS$ = Gros caractères (40 par ligne).

PICA$ = Caractères PICA, normaux (80 par ligne).
ELITE$ = Caractères ELITE (96 par ligne).
COND$ = Caractères condensés (137 par ligne).
ITAL$ = Italiques.
TITRE$ = Caractères gros sur une ligne.
FRAN$ = Caractères France.
USA$ = Caractères USA.

Notre module est présenté dans ce petit programme de démonstration. Remarquez que les lignes Basic sont à la fois courtes et compréhensibles sans manuel. L'écriture fut très rapide et, elle aussi, sans recours au manuel...

```
10 ' EPSON - CODAGE EN CLAIR - M.A 86
15 GOSUB 59000
20 PRINT #8,EFF$;GRAS$;TITRE$;SPC(5);"ES
SAI EPSON"
30 PRINT #8
40 PRINT #8,"ECRITURE EN ";SOUL$;"SOULIG
NANT";SOULX$;" UN MOT"
50 PRINT #8,ELITE$;FRAN$;"PLUS PETITS:Ca
ract}res accentu{es en ELITE";PICA$
60 PRINT #8,EFF$;COND$;"LE MODE CONDENSE
 PEUT PLACER 132 CARACTERES PAR LIGNE";C
ONDX$
70 PRINT #8,GRAS$;"ON PEUT LOGER DE L'";
ITAL$;"ITALIQUE ";ITALX$;"AU MILIEU"
80 PRINT #8,EFF$;"UN MOT EN ";GRAS$;"GRA
S ";GRASX$;"PUIS EN ";GROS$;"GROS";
90 PRINT #8,GROSX$;" CARACT."
100 PRINT #8:PRINT #8,GRAS$;"SANS CHR$ L
E LISTING EST CLAIR,RAPIDE.":PRINT #8,EF
F$
110 END
120 'APRES ESSAI,TAPER DELETE -120 ET SA
VE "EPSON",A
59000 'CODAGES EPSON - M.A. 86
59010 EFF$=CHR$(27)+CHR$(64)
59020 GRAS$=CHR$(27)+"E"
59030 GRASX$=CHR$(27)+"F"
59040 TITRE$=CHR$(14)
59050 ELITE$=CHR$(27)+"M"
```

```
59060 GROS$=CHR$(27)+"W"+CHR$(1)
59070 GROSX$=CHR$(27)+"W"+CHR$(0)
59080 COND$=CHR$(15)
59090 CONDX$=CHR$(18)
59100 PICA$=CHR$(27)+"P"+GROSX$+CONDX$
59110 SOUL$=CHR$(27)+"-"+CHR$(1)
59120 SOULX$=CHR$(27)+"-"+CHR$(0)
59130 ITAL$=CHR$(27)+"4"
59140 ITALX$=CHR$(27)+"5"
59150 FRAN$=CHR$(27)+"R"+CHR$(1)
59160 USA$=CHR$(27)+"R"+CHR$(0)
59400 RETURN
```

CELA DONNE :

**ESSAI EPSON**

ECRITURE EN <u>SOULIGNANT</u> UN MOT
PLUS PETITS:Caractères accentuées en ELITE
LE MODE CONDENSE PEUT PLACER 132 CARACTERES PAR LIGNE
ON PEUT LOGER DE L'*ITALIQUE* AU MILIEU
UN MOT EN **GRAS** PUIS EN GROS CARACT.

SANS CHR$ LE LISTING EST CLAIR,RAPIDE.

Comme il est dit en ligne 120, seul le module en 59000 est à sauvegarder.
Nous avons laissé de la place entre les lignes 59160 et 59400, afin que vous puissiez compléter par des codes que vous utilisez quelquefois.

# Chapitre III

# MODIFICATIONS DES PROGRAMMES ET FICHIERS

Modifier un programme Basic, c'est très facile après LIST ou EDIT, mais encore faut-il connaître le ou les numéros de lignes à modifier, et lorsque le listing est très long, la recherche peut être longue et incomplète. Idem pour un fichier ASCII enregistré par un programme : la modification de nombreuses fiches peut être fastidieuse ou très délicate. Deux exemples :

- Un long programme Basic donne des résultats incohérents parce que vous vous rendez compte que vous avez dû utiliser le même nom de variable pour désigner deux choses différentes, un bug classique, n'est-ce pas ? Quel pointage fastidieux et peu sûr si le listing fait cinq pages d'imprimante ! Et si vous n'avez pas d'imprimante, c'est encore pire !
- Dans un fichier d'adresses, un nom de ville revient très fréquemment et vous vous apercevez que vous l'avez mal orthographié, par exemple, ''VILLEURBANNE'' avec un seul N au lieu de deux. Votre logiciel a bien l'option ''Modification'', mais faire 82 fois cette même retouche, alors là, bon courage ! Même problème dans un texte écrit par ''AMLETTRE'' (traitement de texte AMSOFT) où il faudrait remplacer un nom fréquent par un autre, car ce logiciel n'a pas la fonction remplacement.

En somme, il y a deux problèmes distincts mais souvent combinés : la recherche, le remplacement.
Afin d'automatiser ces longues corvées, nous vous proposons un petit programme Basic et, pour ceux qui possèdent un lecteur de disquettes, l'art et la manière d'utiliser le programme ED.COM du CPM.
S'il s'agit d'un programme Basic, il faut tout d'abord en enregistrer une

version en ASCII sur laquelle nous travaillerons. Par exemple, pour traiter le programme KEY :

SAVE''KEYA'',A

Remarquez le A ajouté à la fin du nom afin d'ôter toute ambiguïté avec la version en Basic. Après modification et sauvegarde de KEYA modifié, on essayera par RUN''KEYA''. Puis, en RAM, on pourra le sauvegarder en Basic par SAVE''KEY2''.

## LE PROGRAMME FOUINE

Une fois lancé, il vous demande s'il faut traiter un fichier ou un programme en ASCII. Dans ce dernier cas, les enregistrements ne seront pas numérotés à l'écran ou sur imprimante. Puis, si vous voulez afficher sur l'écran ou sur imprimante et enfin le nom du fichier ou programme.
Deuxième phase : on demande le mot à rechercher. Il va lire les enregistrements un par un et, s'il y trouve le mot en question, il affiche (écran ou imprimante) le numéro de l'enregistrement et son contenu. En fin de fichier, il indique le nombre total ''N'' d'enregistrements. Trois options sont alors proposées : autre recherche de mot, corrections, quitter.
NOTE : Si au ''mot à rechercher'' vous tapez ENTER, tout le fichier sera listé.
La troisième phase est la correction, elle ne concerne pas les programmes Basic, c'est plus facile par EDIT. D'autre part, elle ne peut s'appliquer qu'aux enregistrements de ''chaînes'' et ne contenant pas de virgules.
C'est le cas le plus courant, et prévoir les cas où il faudrait LINE INPUT au lieu de INPUT et WRITE au lieu de PRINT aurait trop alourdi ce programme.
Le fichier est alors chargé en RAM sous forme de DIM F$(N). Puis, on vous demande le numéro à corriger. Le contenu de cet enregistrement est affiché à l'écran et vous pouvez alors taper sa valeur (chaîne) corrigée. Si vous ne voulez pas corriger (erreur de numéro), tapez deux fois le signe égal. En fin de corrections, tapez Q en guise de numéro.
Quatrième phase : la sauvegarde du fichier corrigé. On demande le nom ; en répondant par ENTER, ce sera l'ancien nom. En fin de sauvegarde, CLS, END.

### *QUELQUES REMARQUES*

- Le DIM bidon de la ligne 100 est pour permettre le ERASE F$ de la ligne 310 ; ce dernier est une précaution au cas où vous relanceriez le programme par un GOTO 120 ou GOTO 300...
- Les lignes 50000 ne sont pas à taper, faites MERGE''MENU'' (chapitre II).

• Ce programme est applicable aux cassettes ou disquettes. S'il permet des recherches globales, il ne permet que des corrections ponctuelles.
• Ce programme permet aussi d'éditer le contenu d'un fichier ASCII et avec numérotation. Très utile pour ceux qui n'ont pas de lecteur de disquettes, donc ne disposent pas des commandes TYPE et ED du DOS CPM.
• Ces corrections sont parfois plus rapides que par le logiciel qui avait créé le fichier... (allusion à certains logiciels du commerce).

```
10 ' FOUINE / EXPLORE UN FICHIER ASCII
20 ' AMSTRAD CPC / M.ARCHAMBAULT 2-86
30 CLS:OPENOUT"BIDON":MEMORY HIMEM-1:CLO
SEOUT
40 LOCATE 14,4:PEN 3:PRINT "F O U I N E"
:PEN 1
50 LOCATE 12,6:PRINT "cherche en ASCII":
PEN 2
60 LOCATE 9,10:PRINT "Programme ou Fichi
er ?":TEX$="PF":GOSUB 50000
70 Q$=MID$(TEX$,K,1)
80 LOCATE 7,14:PRINT "sur Ecran ou Impri
mante ?":TEX$="EI":GOSUB 50000
90 IF K=2 THEN D=8 ELSE D=0
100 DIM F$(12):'DIM BIDON pour ligne 310
110 LOCATE 3,18:INPUT"NOM du Fichier ou
Programme: ",FICH$
120 CLS:N=0
130 PEN 3:LINE INPUT"MOT A RECHERCHER :
",X$:PRINT:PEN 1
140 OPENIN FICH$
150 WHILE NOT EOF:N=N+1
160 LINE INPUT#9,A$
170 IF Q$="F" AND INSTR(A$,X$) THEN PEN
2:PRINT #D,N;": ";:PEN 1:PRINT #D,A$:GOT
O 190
180 IF INSTR(A$,X$) THEN PRINT #D,A$
190 WEND
200 CLOSEIN
210 PRINT:PEN 2:PRINT N;:PEN 1:PRINT "=
FIN :RECHERCHE,CORRIGE,QUITTE ?":TEX$="R
CQ":PEN 3:GOSUB 50000:PEN 1
220 ON K GOTO 120,300,600
300 ' CORRECTION
```

```
310 CLS:LOCATE 3,10:PRINT "Je charge le
Fichier en DIM F$(";N;")":ERASE F$
320 DIM F$(N)
330 OPENIN FICH$:FOR I=1 TO N
340 INPUT#9,F$(I):NEXT:CLOSEIN
350 CLS:LOCATE 1,2:INPUT "CORRIGER QUEL
NUMERO ? (Q=Quitter) ",J$
360 IF UPPER$(J$)="Q" THEN 500
370 J=VAL(J$):IF J=0 OR J>N THEN PRINT C
HR$(7):GOTO 350
380 LOCATE 4,4:PRINT "( Si pas de Correc
tion tapez == )"
390 PEN 2:PRINT:PRINT F$(J):PEN 1
400 PRINT:LINE INPUT A$:IF A$="==" THEN
420
410 F$(J)=A$
420 GOTO 350
500 'SAUVEGARDE
510 CLS:LOCATE 6,4:PEN 3:PRINT "SAUVEGAR
DE DU FICHIER CORRIGE"
520 LOCATE 5,12:PEN 2:PRINT "( ENTER= ME
ME NOM ; Q=QUITTER )":PEN 1
530 LOCATE 10,10:INPUT "NOM du FICHIER :
 ",FICOR$
540 IF FICOR$="" THEN FICOR$=FICH$
550 IF UPPER$(FICOR$)="Q" THEN 600
560 OPENOUT FICOR$:FOR I=1 TO N
570 WRITE#9,F$(I):NEXT:CLOSEOUT
600 CLS:PEN 1
610 END
50000 'REPONSE A UN MENU
50010 LT=LEN(TEX$)
50020 LOCATE 15-LT,24:PRINT"Reponse (";
50030 FOR I=1 TO LT-1
50040 PRINT MID$(TEX$,I,1);",";:NEXT
50050 PRINT RIGHT$(TEX$,1);")";CHR$(154)
;CHR$(243);CHR$(207)
50060 TEX$=UPPER$(TEX$)
50070 R$="":WHILE R$="":R$=INKEY$:WEND
50080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
50090 IF K=0 THEN PRINT CHR$(7);:GOTO 50
070
50100 RETURN
```

**L'UTILISATION DE ED.COM**
**(sur la disquette CPM)**

Ce qui suit ne concerne que les possesseurs de lecteur de disquettes. Un des programmes de la disquette CPM s'appelle ED.COM (ED venant de EDiteur). Il est stupéfiant que la micro-notice AMSTRAD ne parle pas de cette petite merveille. En effet, ce programme va charger en RAM un fichier, ou un programme Basic ou ASCII, afin de le triturer à notre guise et de le sauvegarder ensuite. Outre la recherche de mot et les modifications ponctuelles, on peut lui demander de remplacer un mot par un autre dans tout le fichier, ou à partir de tel endroit, et c'est quasi instantané ! Une bête à tout faire.
Pour bien comprendre et retenir notre démonstration, il est indispensable, avant d'aller plus loin, que vous vous mettiez au clavier de votre AMSTRAD afin d'effectuer point par point ce que nous indiquerons.
Il nous faut un fichier de travail, par exemple, le programme KEY (ou un autre) du chapitre II.

LOAD''KEY'' puis SAVE''KEYA'',A

Sur la *même face* de disquette, il faut maintenant une copie de ED.COM. Mettez en place la disquette CPM, tapez ICPM puis FILECOPY ED.COM, et suivez les indications à l'écran. ED.COM et KEYA sont à présent sur la même face de la disquette de travail. On est toujours sous CPM, la disquette de travail est dans le drive, non protégée en écriture.
Tapez ED KEYA (pas de point) ; en-dessous apparaît le ''curseur'' de ED, à savoir deux points-étoile (:*).
C'est le moment de vous indiquer les diverses commandes de ED. C'est toujours une seule lettre ou signe.

| | | |
|---|---|---|
| # | = | toutes les fiches |
| A | = | APPEND = ajouter |
| F | = | FIND = trouver |
| T | = | TYPE = écrire |
| I | = | INSERT = insérer |
| S | = | SWAP = échanger |
| K | = | KILL = supprimer |
| B | = | BEGINNING = début |
| E | = | END = fin avec sauvegarde + retour à CPM |
| Q | = | QUITTER (sans sauvegarde) + retour à CPM |
| H | = | HOLD = sauvegarde de sécurité. On reste dans ED |
| O | = | Oublie les modifications depuis le dernier H |
| : | = | Numéro. Exemple 15: = fiche n° 15 |
| ↑Z | = | CTRL + Z = séparateur (idem : en Basic) |

Première chose à ne jamais oublier, dire à ED sur quelle partie du fichier on veut opérer ; ici tout le fichier, donc un APPEND total.
Tapez pour cela # A (= ajoute tout). On a alors à l'écran 1:* signifiant que le ''pointeur'' est sur la fiche n° 1.

Visionnons les dix premières fiches ; pour cela tapons 10T. Nous retrouvons alors les dix premières lignes de notre programme Basic, mais, en plus, numérotées de 1 à 10 (ce que n'aurait pas fait TYPE KEYA en CPM direct). Si nous avions tapé #T, nous aurions ''listé'' tout le fichier. OK ?
Remarquez que nous avons de nouveau 1:*. Déplaçons ce pointeur en tapant 8:, puis encore 10T. Nous listons alors les fiches n° 8 à 17, et le pointeur est resté à 8.
Pour le ramener à 1, on a le choix entre 1: ou plus simplement B. Essayez B.
Listons la fiche n° 23 toute seule, faisons 23: puis 1T. A noter que T tout seul a le même effet.
Maintenant que nous savons nous promener dans le fichier, abordons des choses plus intéressantes :

*LA COMMANDE FIND*

Ramenons le pointeur à 1 en tapant B, et localisons un mot, par exemple ''PLOT'' parce que figurant dans ce fichier. C'est tout simple, tapez FPLOT (pas d'espace entre le F et le mot à chercher). L'écran affiche 33:* parce qu'il a trouvé en fiche n° 33, mais attention, le fameux pointeur n'est pas au début de la fiche 33, mais dedans, tout de suite après PLOT. La preuve, si vous tapez T, l'écran affiche *ce qui suit* le mot PLOT dans la fiche 33. Essayez.
Voilà l'astuce contre cela. Refaites B, puis FPLOT, puis tapez le *chiffre zéro.* Vous avez alors la fiche 33 au complet.
Repartons, tapons B puis FPLOT, encore FPLOT. On a le message :

BREAK ''#''AT
33:*

Cela veut dire qu'il a tout (#) exploré, annulé votre commande et qu'il revient à 33. Parce qu'il n'y avait plus d'autre PLOT.
Retour par B puis 3FLOCA. Réponse 25:*, c'est le *troisième* LOCATE. #FLOCA aurait indiqué le dernier.
Vous voyez que la commande F n'est pas répétitive, il faut la retaper chaque fois (CPM 2.2). Donc, sur ce point le programme FOUINE était plus intéressant puisqu'il affiche *toutes* les ''occurences''.

*LA COMMANDE SWAP (Echange)*

Elle est la plus utile et aussi la plus spectaculaire. Retour en B.

2SPRINT↑ZWRITE (pas d'espaces)

signifie ''Les deux premières fois que tu rencontres PRINT, tu remplaces par WRITE''. Le nombre placé devant une ''lettre-commande'' veut dire le nombre de fois. Par défaut, c'est 1, le # voulant dire maxi.

Important : le séparateur ↑Z s'obtient par CTRL et Z. Si vous tapez la touche ''↑'' (à gauche de CLR), puis la touche Z, c'est la même chose à l'écran, mais ED ne comprend pas.
Tapez 10: puis

#SKEY↑ZTOUCHE

puis B et 20T et admirez le travail !
Pour annuler ce massacre, tapez la lettre O. Confirmez par Y (YES), KEYA version d'origine est rechargée. N'oubliez pas le #A...
Tapez #S'↑ZREM ↑Z (un espace après REM, confirmé par le second ↑Z).
Vous devinez ce qu'il vient de faire.
Sauvegarde de sécurité : pour cela tapez H. Ceci fait, vous avez : * sans le 1. Refaites #A, puis 20T, pour voir...
Puissant et rapide la bête !
Si au lieu de taper H, on avait tapé E (END), il y aurait eu sauvegarde mais avec, ensuite, retour à CPM (A>).

*LES AUTRES COMMANDES UTILES*

Ce sont K (KILL) et I (INSERT). Tapez 3: puis K. Le fiche n° 3 est détruite et les suivantes sont renumérotées. Pour vous en assurer, faites B puis 5T.
Retapez 3: puis

130 REM LIGNE REMISE

Réponse * tout seul. Faites deux fois ENTER puis B et 5T.

**RESUME** : Pour modifier le contenu d'une fiche :
1 — Appelez son numéro (suivi de :)
2 — Tapez K pour la détruire
3 — Tapez I suivi du nouveau contenu
4 — Deux fois ENTER

Tapez Q (quitter sans sauvegarde) ; après confirmation par Y, vous quittez ED et vous voilà revenu à CPM. Tapez alors :

TYPE KEYA

C'est le ''listing'' sauvegardé par le H de ED (REM au lieu de '). Si vous préférez cette version au KEY original, revenez au Basic par CTRL + SHIFT + ESC, et faites :

LOAD''KEYA'' puis SAVE''KEY''

A l'occasion, faites RUN''DISCUT'' (chapitre II) pour faire le ménage en effaçant les .BAK et KEYA qui ne servent plus à rien.

## CONCLUSION

L'auteur s'excuse de cette ''séance de travaux pratiques'' sur ED.COM, mais il n'y avait pas d'autres méthodes pour vous faire découvrir ce *mini langage* super utile. Les ouvrages traitant du CPM avec clarté étant rarissimes.
ED.COM a d'autres commandes mais, à notre avis, de moindre intérêt.

FOUINE ou ED, que choisir ? Pour rechercher *toutes* les occurences d'un mot, par exemple tous les ''GOTO 1000'', FOUINE est préférable, mais il est surtout un localisateur. Pour les modifications, ED.COM est le champion.

Nous avons souvent parlé de ''disquette de travail'' : c'est une sorte d'établi où le programme est simplement mis au point. Une fois terminé, il sera transféré ailleurs et effacé de la disquette de travail. Sur celle-ci, ayez en permanence les programmes :

KEY.BAS
DISCUT.BAS
FOUINE.BAS
ED.COM

ce qui ne représente que 13 kilo-octets sur les 159 d'une face de disquette. Mais quel gain de temps !

# Chapitre IV

# COPIE DE PROGRAMMES EN LANGAGE MACHINE

Soyons très clairs dès le départ. Il n'est pas question dans ce chapitre de vous apprendre à programmer en langage machine ; un ouvrage d'épaisseur double y suffirait à peine ; mais de vous permettre d'utiliser des programmes écrits par d'autres, et publiés dans des revues ou des livres. Nuance !
Pour le débutant, ces listings en plusieurs colonnes sont déroutants, or le titre et le rôle de tels programmes sont souvent très alléchants. Comment les entrer au clavier sans maîtriser cette technique ?
Nous avons créé un programme Basic pour cela ; il se chargera de tout, très rapidement, tout en surveillant vos erreurs. Toutefois, afin de ne pas ''taper idiot'', il est absolument indispensable de posséder quelques notions élémentaires sur le langage machine. Nous serons à la fois clairs et brefs ! Ces ''notions triées'' sont très superficielles, faciles à comprendre, peu nombreuses, mais, répétons-le, indispensables.

## QU'EST-CE QU'UN PROGRAMME EN LANGAGE MACHINE ?

C'est une suite continue de nombres, de 0 à 255, qui ont été logés quelque part dans la mémoire. Par exemple, de *l'adresse* 32901 à 32984 ; sa *longueur* est donc de 84 octets (et non pas 83...) et son *adresse départ* est 32901. Pour lancer ce programme à partir du Basic, ce n'est pas RUN, mais CALL 32901. Ne vous occupez pas de la longueur, il saura où s'arrêter.
A quoi correspondent ces nombres de 0 à 255 (1 octet) ? A une commande codée, *suivie* par autant de valeurs numériques qu'elle exige. Ceci fait, la machine sait que le nombre suivant est un numéro de commande, et ainsi de suite.

Donc, le ''vrai listing'' de ce programme est une suite de 84 nombres, de 0 à 255, qui ont été ''pokés'' à partir de l'adresse 32901 (dans notre exemple). Pas très lisible cela ! Rassurez-vous, le programmeur ne les a pas entrés directement, ce serait super maso ! Il s'est servi d'un logiciel appelé ''Assembleur''. Grâce à lui, il a tapé son programme ''en clair'' comme vous l'auriez fait en Basic, c'est-à-dire qu'il dispose de commandes (mots de deux à quatre lettres) qu'il fait suivre des paramètres (valeurs) qu'il désire. En somme un autre langage que le Basic, que l'on appelle de l'Assembleur. La machine ne comprend pas du tout ce langage, mais le logiciel, oui. C'est lui qui le traduit en cette suite de nombres et les ''poke'' en mémoire (c'est l'assemblage).
Ceci fait, le programmeur l'a sauvegardé sur cassette ou disquette, en lui donnant un nom, puis il a demandé à son logiciel assembleur de le lister sur imprimante. Nous allons revenir sur ces listings assez spéciaux, mais avant cela, disons les avantages des programmes en langage machine :

- C'est beaucoup plus rapide que l'équivalent en Basic, de 3 à 50 fois selon le cas. C'est pourquoi les logiciels de jeux (animation) sont le plus souvent en ce langage (on dit aussi programmes ''binaires'').
- On peut faire des choses impossibles avec le Basic livré avec l'ordinateur. Plus ponctuellement, on peut créer des fonctions Basic manquantes, que l'on appelle des ''RSX''. A noter que ces courts programmes ne sont pas lancés par CALL, mais par barre verticale, suivie du nom de baptême (inclus dans le programme), par exemple :

  ICERCLE 220,350,40,3

  va tracer un cercle de rayon 40 en PEN 3 avec le centre situé aux coordonnées 220,350. La littérature abonde de listings de RSX et certains sont très alléchants !

Est-ce qu'un programme en langage machine tient moins de place en mémoire que son homologue en Basic ? La plupart du temps oui, mais ce n'est pas très spectaculaire... Disons que c'est là un atout faible, parfois inexistant.

## LE LISTING EN ASSEMBLEUR

Ne soyez plus de ceux qui appellent cela des''pages en chinois''...
Toute la partie droite restitue ce que le programmeur a tapé en Assembleur : *on s'en moque*, mais disons quand même à quoi cela correspond.
La colonne de droite sont des REM.
Celle plus à gauche est son programme, ces mots de 2 ou 4 lettres (LD, ADD, JP, etc.) appelés ''mnémoniques'' suivis de paramètres ou de noms de sous-programmes (équivalents de nos GOTO ou GOSUB), des noms à lui.
Encore plus à gauche, la colonne des étiquettes (ou labels) ; ce sont les noms de ces sous-programmes, car il n'y a pas de numéros de ligne comme en Basic.

Encore plus à gauche, la colonne de ces nombres établis par le logiciel assembleur. C'est uniquement cette colonne qui nous intéresse. Ces nombres sont écrits en hexadécimal, on y reviendra.
Encore plus à gauche (et c'est la dernière), la colonne des adresses où sont pokés ces nombres. On n'a pas à la retaper, mais elle nous sera utile pour vérifier que l'on ne s'est pas trompé. En fait, seule la première adresse nous intéresse, c'est la fameuse ''ADRESSE DEPART''
A noter à propos de cette dernière : ou bien le programmeur l'a choisie *arbitrairement*, ou il a confié ce soin au logiciel assembleur qui, si le programme est court, opte généralement pour &A000 (=40960 en décimal) parce que ''A000'', c'est facile à retenir...
Les listings publiés dans les livres ou mensuels sont les photos des listings obtenus avec tel Assembleur, d'où certaines différences de présentation : aucune importance. Certains débutent par plusieurs lignes d'en-tête qui présentent les divers sous-programmes ; n'en tenez pas compte.

Revenons à la fameuse colonne de nombres qui nous intéresse, c'est généralement la deuxième en partant de la gauche ; elle contient les valeurs en hexadécimal que nous allons entrer au clavier, octet par octet, c'est-à-dire *deux caractères* par octet : chiffres de 0 à 9 et lettres de A à F. Une ligne de cette colonne peut compter six caractères, exemple D9A782 : il faudra les entrer séparément *par groupes de deux*, à savoir les trois octets D9, A7 et 82 (toujours en hexa). C'est super important.

## LA NOTATION HEXADECIMALE
## (Programme HEXADEC)

Le préfixe ''hexadéci'' signifie seize, car nous disposons de 16 caractères, dix chiffres plus six lettres de A à F. Ainsi, deux de ces caractères permettent 256 combinaisons, donc de compter de 0 à 255 (00=0 et &FF=255). Le très court programme Basic ''HEXADEC'' permet de faire la traduction hexadécimal/décimal dans les *deux* sens, et ce entre 0 et 65535 (à partir de 256 et jusqu'à 65535, il faut deux octets).
Sans être indispensable, ce petit programme est souvent très utile. Pour changer de sens dans la traduction, il vous suffira de taper la lettre Q (=quitter). En Hexa, inutile de taper le ''&''. Si cette routine a été ''mergée'' à un programme en cours de mise au point, appelez-la par GOTO 40000, et touche ESC pour en sortir. Mieux, programmez la touche ''6'' du pavé numérique par :

```
KEY 134,"GOTO 40000"+CHR$(13)
```

Le début n'a pas de CLS, c'est voulu.

```
40000 ' HEXADEC - CONVERSION HEXA <-->
DECI - M.ARCHAMBAULT
40010 PRINT STRING$(40,"-");:PRINT "   Q
 pour Transitions HEXA <--> DECI"
40020 PRINT:INPUT "  &",H$:H$=UPPER$(H$)
:IF H$="Q" THEN 40100
40030 FE=0:FOR I=1 TO LEN(H$):IF ASC(MID
$(H$,I,1)) >70 OR ASC(MID$(H$,I,1)) <48
THEN FE=1
40040 NEXT:IF FE THEN 40080
40050 IF LEN(H$)<=2 THEN PRINT "  ";VAL(
"&"+H$):GOTO 40090
40060 IF LEN(H$)=3 THEN H$="0"+H$
40070 IF LEN(H$)=4 THEN D=VAL("&"+LEFT$(
H$,2))*256+VAL("&"+RIGHT$(H$,2)):PRINT "
   ";D:GOTO 40090
40080 PRINT CHR$(7)
40090 GOTO 40020
40100 PRINT: INPUT "DECI : ",D$
40110 IF UPPER$(D$)="Q" THEN 40020
40120 FE=0:FOR I=1 TO LEN(D$):IF ASC(MID
$(D$,I,1)) >57 OR ASC(MID$(D$,I,1)) <48
THEN FE=1
40130 NEXT
40140 D=VAL(D$):IF FE OR D>65535 THEN PR
INT CHR$(7);:GOTO 40100
40150 PRINT"         &"+HEX$(D)
40160 GOTO 40100
40170 ' ------- FIN DE LISTING -------
```

Après usage, effacez par DELETE 40000-40900.

## LE PROGRAMME MACHDIM

Le travail est simple : on entre l'adresse départ puis chacun des octets du listing (colonne n° 2) est entré, puis POKE à l'adresse à chaque fois augmentée de 1. Trois lignes de Basic suffiraient, or notre programme est bien plus long ! Pourquoi ?
Parce que non seulement l'erreur est humaine, mais, ici, elle est super facile et serait très fastidieuse à corriger. Il nous faut un produit bourré de sécurités anti-étourderies, des ''sorties de secours'', des possibilités de contrôle et de correction rapides et faciles. Ce confort, ce super luxe, est en fait quasi obligatoire, car taper des centaines de nombres

en hexa, sans rien y comprendre, n'a vraiment rien de passionnant ; le risque d'erreur est alors énorme car l'attention se relâche, le Basic doit veiller à cela.
En voici le principe : les valeurs hexa entrées et leurs adresses correspondantes sont mises en tableaux DIM, on ne les POKE pas encore.
En fin de saisie, ces tableaux sont enregistrés sur cassette ou disquette.
Seconde phase : on lance un second programme Basic, très court celui-là, ''POKEDIM'' qui va recharger ces valeurs tout en effectuant le ''pokage'' en mémoire. On peut alors sauvegarder le précieux programme en langage machine puis l'essayer. Cette méthode très indirecte pour mettre en RAM le programme binaire peut surprendre : c'est en fait une sécurité anti-plantages pour des programmes longs ou logés très haut ou très bas dans la mémoire. Fiabilité avant tout.

Revenons à MACHDIM. De quoi disposons-nous ?
- D'un écran de saisie en plusieurs WINDOWS, très clair et sans équivoque ;
- d'une surveillance permanente des octets en hexa (par Basic) ;
- d'un contrôle visuel des concordances adresses/valeurs ;
- de la possibilité d'interrompre la saisie pour sauvegarde partielle, visionner, corriger telle valeur ;
- de la possibilité de poursuivre la saisie un autre jour ;
- des adresses présentées en décimal et en hexadécimal ;
- de la possibilité de lister la saisie sur écran ou sur imprimante (pour vérifications) ;
- en fin de saisie, et *après sauvegarde* des DIM (précaution élémentaire), de la possibilité d'essayer le programme ; c'est un pokage des valeurs, suivi d'un CALL ;
- en quittant la saisie par ''Q'', de l'accès aux diverses options lesquelles sont légendées dans un WINDOW ;
- chaque option choisie demande confirmation ;
- confirmation et paramètres pour une option sont dans un autre WINDOW ;
- d'autres WINDOWS affichent en permanence les légendes des colonnes et la fameuse adresse départ, en hexa et en décimal.

**NOTE** : Comme vous le constatez, l'auteur aime son petit confort. Et vous ?

## LE MODE D'EMPLOI

L'écran vous demande tout d'abord d'entrer l'adresse départ, en décimal ou en hexa. Dans ce dernier cas, après le ''&'' pas d'espace et obligatoirement quatre caractères, exemple &9800.
Apparaît alors l'écran de saisie, multi-WINDOW, multicolore (ça réveille...) ; moitié gauche = saisie, moitié droite = les interventions.
La partie saisie est en trois colonnes, adresse en décimal, sa traduction en hexa et les valeurs hexa à entrer ; le curseur vous y attend.

C'est bien sûr le programme qui affiche les adresses. A droite, la fenêtre menu propose : Q = quitter, E = erreur, V = vision, I = impression, L = LOAD, S = SAVE et C = CALL.
La fenêtre de saisie comprend vingt lignes avec scrolling. De temps à autre, vérifiez qu'un oubli n'a pas provoqué un décalage adresse/valeur. Les valeurs entrées sont obligatoirement en *deux* caractères hexa, et en majuscules (sinon BIP sonore !). Ne tapez pas le "&", il est sous-entendu.

```
     ADRESSE       VALEUR     |  ADRESSE DEPART:
   deci      hexa             |  40704    &9F00
------------------------------|------------------------
 40704      9F00      CD      |  Valeurs en DIM
 40705      9F01      A6      |
 40706      9F02      9F      |  Q = Quitter.
 40707      9F03      3E      |  E = Erreur.
 40708      9F04      1B      |  V = Vision.
 40709      9F05      CD      |  I = Impression.
 40710      9F06      9D      |  L = LOAD
 40711      9F07      9F      |  S = SAVE
 40712      9F08      3E      |  C = CALL
 40713      9F09      31      |------------------------
 40714      9F0A      CD      |
 40715      9F0B      9D      |
 40716      9F0C      9F      |
 40717      9F0D      CD      |
 40718      9F0E      BA      |  IMPRESSION:
 40719      9F0F      I       |   ( O/N ) O
```

Si vous n'entrez qu'un seul caractère, le programme consulte s'il est dans la liste des caractères d'intervention du menu.
En vision, il va lister votre saisie à partir de l'adresse de votre choix. Il présente vingt lignes et s'arrête ; pas comme LIST en Basic... Trois nouvelles options apparaissent : S = suite, E = erreur repérée, à corriger et F = fin de vision (= retour à la saisie ou autre option).
Exemple : pour taper le programme "HARD9F00" (Hard-copy d'écran) du chapitre XVI, il vous faudra entrer l'adresse départ &9F00 (ou 40704 en décimal), puis saisir successivement CD, A6, 9F, 3E, 1B, etc.
En fin de saisie, tapez Q, puis S (SAVE) en donnant le nom "COPYDIM". Puis Q pour quitter.
Ceci fait, chargez POKEDIM. Il vous demande l'adresse départ puis le nom du fichier à lire ("COPYDIM"). Le programme binaire est alors poké en RAM. L'écran vous affiche l'adresse départ et la longueur en octets. Notez bien ces valeurs par écrit ; elles seront indispensables plus tard pour enregistrer ce programme binaire sur d'autres cassettes ou disquettes. Pour la sauvegarde en binaire, il vous demande un nom (un autre), disons "HARD9F00".
Remarquez l'astuce qui consiste à inclure l'adresse départ (pour le CALL) dans le nom d'un programme en langage machine.
Si vous avez une imprimante (autre que la DMP-1), essayez-le par CALL &9F00.

## CONCLUSION

Avec ces petits programmes utilitaires en Basic et ces quelques notions générales, vous êtes à présent opérationnel pour profiter (enfin !) des listings en Assembleur ; même sans le pratiquer ni le comprendre ; il ne vous impressionnera plus.

```
10 'MACHDIM :CHARGEUR DE PROGRAMME EN LA
NGAGE MACHINE EN DIM
20 ' AMSTRAD CPC / Michel Archambault 19
85
30 OPENOUT "BIDON":MEMORY HIMEM-1:CLOSEO
UT
40 DEFINT J,K,V:DIM P(3000),V(3000)
50 MODE 1:CLS:BORDER 16
60 PEN 3:LOCATE 6,4:PRINT "MISE EN DIM D
E PROGRAMMES EN"
70 LOCATE 11,6:PRINT "EN LANGAGE MACHINE
."
80 PEN 2:LOCATE 8,9:PRINT "Michel Archam
bault - 1985"
100 PEN 2:LOCATE 2,20:PRINT "Si en HEXAD
ECIMAL faites preceder de &":LOCATE 10,2
2:PRINT "puis QUATRE caracteres."
110 PEN 1:LOCATE 8,15:INPUT "ADRESSE DEP
ART : ",AD$
120 IF LEFT$(AD$,1)="&" THEN AD=VAL("&"+
RIGHT$(AD$,2))+256*VAL(LEFT$(AD$,3)):IF
LEN(AD$)<>5 THEN AD=0:GOTO 110
130 IF AD=0 THEN AD=VAL(AD$)
1000 ' ECRAN DE SAISIE
1010 WINDOW #0,1,22,6,25:CLS:WINDOW #1,1
,22,1,4:PAPER #1,1:PEN #1,0:CLS #1
1020 LOCATE #1,5,2:PRINT #1,"ADRESSE";SP
C(4);"VALEUR"
1030 LOCATE #1,3,3:PRINT #1,"deci";SPC(4
);"hexa"
1040 WINDOW #2,23,40,1,18:PAPER #2,2:PEN
 #2,0:CLS #2
1050 LOCATE #2,2,2:PRINT #2," ADRESSE DE
PART:"
1060 LOCATE #2,3,3:PRINT #2,USING "#####
";AD;:PRINT #2,SPC(3);"&";HEX$(AD)
```

```
1070 PRINT #2,STRING$(18,"_"):PEN#2,3:PR
INT #2,"  Valeurs en DIM":PEN#2,0:PRINT
#2
1080 PRINT #2," Q = Quitter."
1090 PRINT #2," E = Erreur."
1100 PRINT #2," V = Vision."
1110 PRINT #2," I = Impression."
1120 PRINT #2," L = LOAD"
1130 PRINT #2," S = SAVE"
1140 PRINT #2," C = CALL"
1150 OPT$="QEVILSC"
1160 PRINT #2,STRING$(18,"_")
1170 WINDOW #3,23,40,18,40:PAPER #3,2:PE
N #3,3:CLS #3
2000 ' SAISIE
2010 A=AD:J=1
2020 PRINT USING "######";A;:PRINT SPC(4
);RIGHT$("000"+HEX$(A),4);SPC(3);
2030 K=0:INPUT "",V$:IF V$="" OR LEN(V$)
>2 THEN PRINT CHR$(7):GOTO 2020
2040 IF LEN(V$)=1 THEN K=INSTR(OPT$,V$):
IF K=0 THEN PRINT CHR$(7):GOTO 2020
2050 IF K THEN ON K GOTO 3000,4000,5000,
6000,7000,8000,9000
2060 IF LEFT$(V$,1)>"F" OR RIGHT$(V$,1)>
"F" THEN PRINT CHR$(7):GOTO 2020
2070 V=VAL("&"+V$)
2080 P(J)=A:V(J)=V
2090 A=A+1:J=J+1:GOTO 2020
3000 'QUITTER
3010 CLS #3:LOCATE #3,2,3:INPUT #3,"ON A
RRETE (O/N) ",Q$
3020 IF Q$="O" THEN MODE 1:CLS:BORDER 1:
END
3030 IF Q$="N" THEN 3050
3040 PRINT CHR$(7);:GOTO 3010
3050 CLS #3:GOTO 2020
4000 ' ERREUR
4010 CLS #3:LOCATE #3,6,2:PRINT #3,"ERRE
UR:"
4020 LOCATE #3,3,4:PRINT #3,"Adresse DEC
I:"
4030 LOCATE #3,3,6:INPUT #3,"",AE$:AE=VA
```

```
L(AE$):IF AE<AD THEN PRINT CHR$(7);:GOTO
 4090
4040 PRINT USING "######";AE;:PRINT SPC(
4);RIGHT$("000"+HEX$(AE),4);SPC(3);
4050 INPUT "",V$:IF LEN(V$)<>2 THEN PRIN
T CHR$(7);:GOTO 4040
4060 IF LEFT$(V$,1)>"F" OR RIGHT$(V$,1)>
"F" THEN PRINT CHR$(7);:GOTO 4040
4070 V=VAL("&"+V$)
4080 JE=AE-AD+1:V(JE)=V
4090 IF FVIS THEN FVIS=0:GOTO 5000
4100 CLS #3:GOTO 2020
5000 'VISION
5010 CLS#3:LOCATE #3,3,2:INPUT #3,"VISIO
N ?(O/N) ",Q$:IF Q$="N" THEN 5140
5020 IF Q$<>"O" THEN 5010
5030 CLS#3:LOCATE #3,2,2:PRINT #3,"Adres
se DEPART"
5040 LOCATE #3,2,4:INPUT #3,"en DECI:",A
V$:AV=VAL(AV$)
5050 IF AV<AD THEN PRINT CHR$(7);:GOTO 5
000
5060 CLS#3:CLS:FOR I=AV TO AV+19:K=I-AD+
1
5070 PRINT USING "######";P(K);:PRINT SP
C(4);RIGHT$("000"+HEX$(P(K)),4);SPC(3);R
IGHT$("0"+HEX$(V(K)),2):NEXT:AV=AV+20
5080 CLS#3:LOCATE #3,2,2:PRINT #3,"Suite
,Erreur,Fin"
5090 LOCATE #3,5,4:INPUT#3,"( S/E/F ) ",
Q$
5100 IF Q$="S" THEN 5060
5110 IF Q$="F" THEN 5140
5120 IF Q$="E" THEN FVIS=1:GOTO 4000
5130 PRINT CHR$(7);:GOTO 5080
5140 CLS#3:CLS:GOTO 2020
6000 ' IMPRESSION
6010 CLS #3:LOCATE #3,2,3:PRINT #3,"IMPR
ESSION:"
6020 INPUT #3,"  ( O/N ) ",Q$:IF Q$="N"
THEN 6070
6030 IF Q$<>"O" THEN PRINT CHR$(7);:GOTO
 6000
```

```
6040 PRINT #8,CHR$(27);CHR$(64):FOR I=1
TO J-1
6050 PRINT #8,USING "######";P(I);:PRINT
 #8," : ";RIGHT$("000"+HEX$(P(I)),4);" -
 ";RIGHT$("0"+HEX$(V(I)),2)
6060 NEXT
6070 CLS#3:GOTO 2020
7000 'LOAD
7010 CLS#3:LOCATE #3,2,2:INPUT #3,"LOAD
( O/N )",Q$
7020 IF Q$="N" THEN 7120
7030 IF Q$<>"O" THEN 7000
7040 LOCATE #3,2,4:INPUT #3,"NOM:",FICH$
7050 IF FICH$="" THEN PRINT CHR$(7);:GOT
O 7040
7060 FICH$=LEFT$(FICH$,8)
7070 J=0:OPENIN FICH$
7080 J=J+1:IF EOF THEN 7100
7090 INPUT #9,P(J),V(J):GOTO 7080
7100 CLOSEIN
7110 A=P(J-1)+1
7120 CLS#3:GOTO 2020
8000 'SAVE
8010 CLS#3:LOCATE #3,2,2:INPUT #3,"SAVE
( O/N ) ",Q$
8020 IF Q$="N" THEN 8090
8030 IF Q$<>"O" THEN 8000
8040 LOCATE #3,2,4:INPUT #3,"NOM:",FICH$
8050 IF FICH$="" THEN PRINT CHR$(7);:GOT
O 8040
8060 FICH$=LEFT$(FICH$,8)
8070 OPENOUT FICH$
8080 FOR I=1 TO J-1:WRITE#9,P(I),V(I):NE
XT:CLOSEOUT
8090 CLS#3:GOTO 2020
9000 'ESSAI DE CALL
9010 CLS #3:LOCATE #3,2,4:INPUT #3,"ESSA
I CALL (O/N) ",Q$
9020 IF Q$="N" THEN 9050
9030 IF Q$<>"O" THEN PRINT CHR$(7);:GOTO
 9000
9040 FOR I=1 TO J-1:POKE P(I),V(I):NEXT:
CALL AD
```

```
9050 CLS#3:GOTO 2020
9060 '--- FIN DE LISTING ---
```

VARIABLES DE MACHDIM

```
- A : ADRESSE EN COURS
- AD,AD$ : ADRESSE DEPART
- AE,AE$ : ADRESSE DE L'ERREUR
- AF,AF$ : ADRESSE FINALE
- AV,AV$ : ADRESSE DEPART VISION
- DIM P(3000) : TABLEAU DES ADRESSES
- DIM V(3000) : TABLEAU DES VALEURS V
- FICH$ : NOM DU FICHIER EN ASCII
- FVIS : FLAG DE VISION
- I,J,K : INDICES DE COMPTAGES
- JE : INDICE DE L'ERREUR
- LG : LONGUEUR DU FICHIER
- Q$ : REPONSE A UNE QUESTION
- V,V$ : VALEUR DE L'OCTET SAISI
```

A présent le programme final : POKEDIM.

```
10 'POKEDIM:POKAGE D'UN FICHIER ASCII CR
EE PAR MACHDIM
20 'AMSTRAD CPC / M.Archambault 1985
30 OPENOUT "BIDON":MEMORY HIMEM-1:CLOSEO
UT
40 CLS:INPUT"ADRESSE DEPART (& ou deci)
",AD$:IF LEFT$(AD$,1)="&" THEN AD=VAL(LE
FT$(AD$,3))*256+VAL("&"+RIGHT$(AD$,2)):G
OTO 60
50 AD=VAL(AD$)
60 IF AD<1 THEN 40 ELSE IF AD<HIMEM THEN
 MEMORY AD-1
70 CLS:INPUT"NOM DU FICHIER:",FICH$
80 OPENIN FICH$:J=0
90 J=J+1:IF EOF THEN 120
100 INPUT#9,A,V:IF J=1 THEN AD=A
110 POKE A,V:GOTO 90
```

```
120 CLOSEIN
130 L=J-1
140 PRINT "NOTEZ pour ";FICH$;" :"
150 PRINT "    Adresse DEPART=";AD
160 PRINT "           Longueur=";L
170 PRINT:INPUT " SAVE en BINAIRE ? (O/N
) ",Q$:Q$=UPPER$(Q$)
180 IF Q$="O" THEN 200
190 IF Q$<>"N" THEN 170 ELSE END
200 PRINT:INPUT"NOM DU FICHIER(ou ENTER)
:",NOM$
210 IF NOM$="" THEN NOM$=FICH$
220 SAVE NOM$,B,AD,L:END
230 '---FIN DU LISTING-----
```

# Chapitre V

# LES CARACTERES GRAPHIQUES

On est quelquefois amené à créer des caractères spéciaux qui n'existent pas sur l'AMSTRAD, lettre grecques, symboles liés à une certaine science, quand il ne s'agit pas d'éléments graphiques à juxtaposer pour faire un ''logo tape-à-l'œil'', ou une petite animation simple en mode texte (voir MPSA chapitre XIV). Certains linguistes ont même redéfini toutes les touches afin de pouvoir écrire en langues non romaines (grec, hébreux, russe, idéogrammes, etc.).
Définir un caractère et son codage pour la commande SYMBOL n'est pas très compliqué (MPSA page 76), mais s'il y en a beaucoup, cela devient vite fastidieux. Un exemple : les six caractères français de la routine ''AZERTY'' du chapitre II de ce livre.
Le programme ''DEFCAR'' que nous vous proposons a été conçu pour gens pressés et n'aimant faire aucun calcul.
Le second problème, rarement abordé, est de reproduire ces motifs sur imprimante ! DEFCAR le fait aussi et tout aussi automatiquement.
Quant au tracé de votre caractère spécial, vous le dessinez et corrigez à l'écran à l'aide du joystick.

## RAPPEL SUR LE CARACTERE GRAPHIQUE

Chaque caractère de l'AMSTRAD (lettres, etc.) est défini par une grille de huit carreaux sur huit. Les carreaux couleur PEN ont la valeur 1, les autres zéro.
Chacune des huit rangées horizontales de notre caractère est donc une suite de 1 et 0. Considérons que c'est un nombre en écriture binaire, que l'ordinateur sait bien sûr convertir en décimal et vice-versa. Chaque rangée horizontale est définie par un nombre. Le nouveau caractère sera mis en mémoire par l'instruction SYMBOL, suivi du code ASCII choisi et de ces huit nombres.
Pour reproduire ce motif à l'écran, le CPC va dessiner ces huit images binaires les unes sous les autres.

Que ceux qui sont un peu ''perdus'' fassent ce petit essai :

PRINT BIN$(102,8)

on obtient 01100110
A présent, l'opération inverse :

PRINT &X 01100110

on a bien 102.

## LE CAS DE L'IMPRIMANTE

Contrairement à l'écran qui trace ses petits points (''pixels'') de gauche à droite, puis de haut en bas, la tête d'impression de notre imprimante frappe des colonnes verticales de points, en se déplaçant de gauche à droite !
Il va donc falloir envoyer à l'imprimante les ''images binaires'' des huit colonnes *verticales* de notre caractère, de gauche à droite. En somme, c'est le même principe, mais dans l'autre sens.
Pour imprimer des caractères usuels (A, B, C, etc.), l'imprimante à ces codages dans sa ROM ; on lui envoie le code ASCII et elle se charge de dessiner le caractère. Il faut ici la prévenir qu'il ne s'agit pas de codes ASCII, mais d'images binaires, c'est la commande ''Bit image'' (que ne possédait pas la DMP 1) : CHR$(27);''K''.
La syntaxe de cette commande est assez lourde. Pour *chaque* caractère à imprimer, il faudra écrire :

PRINT #CHR$(27);''K'';CHR$(8);CHR$(0)

suivi de nos huit nombres ''dans'' des CHR$. Une terrible corvée d'écriture ? Mais non, si l'on utilise quelques astuces que nous décrirons plus loin.

## LE PROGRAMME DEFCAR

Avant de le décortiquer, voyons d'abord le mode d'emploi.
— On vous demande si l'imprimante est prête. Si vous n'en possédez pas, répondez bien sûr ''N''.
— Dans l'angle en haut à gauche apparaît une grille 8 × 8. Dans l'un des 64 carreaux clignote un point rouge ; c'est notre curseur, que vous déplacez à l'aide du joystick dans ses huit directions.
— Pour ''noircir'' un carreau, c'est FIRE.
— Pour effacer un carreau, c'est la barre d'espacement.
— Lorsque votre dessin est terminé, tapez la lettre ''Q'' pour quitter.
— On vous demande alors à quel caractère du clavier vous voulez l'attribuer. Vous pouvez *indifféremment* répondre par le caractère ou par le code ASCII (de 32 à 255).
— Cas sans imprimante : l'écran vous demande de noter les paramè-

tres qui devront suivre la commande SYMBOL. Avec imprimante, ces résultats sont seulement imprimés.
- A l'écran apparaît le code ASCII du caractère demandé et sa nouvelle représentation, grandeur nature cette fois.
- Avec imprimante, le nouveau caractère est imprimé, ainsi que les huit paramètres pour imprimante (codage vertical). Trait de séparation.
- L'écran vous demande si vous voulez redéfinir d'autres caractères. Si vous répondez non, CLS et l'écran vous conseille de faire un CTRL + SHIFT + ESC (ou un CALL 0), car les touches définies restent actives même après un RUN !

**NOTE** : En option imprimante, il apparaît à l'écran un astérisque rouge en face de la rangée supérieure de la grille. Cela signifie ''interdite'' car elle ne sera pas reproduite sur papier. En effet, l'interface imprimante de nos CPC est malheureusement une 7 bits et non une 8 bits (par mesure d'économie du constructeur).

**RESUMONS** : On dessine avec le joystick et on a tout de suite après les résultats graphiques en vraie grandeur sur écran, sur imprimante, avec tous les paramètres calculés pour les transcrire dans vos programmes.

A présent, voyons quelques détails Basic :
Ligne 30 : Les codes ASCII redéfinissables seront égaux ou supérieurs à 32 (barre d'espacement).
- Ligne 60 : FIMP est un ''flag'' qui est à 1 si l'imprimante est prête.
- Module 600 (grille) : Nous utilisons le ''mode transparent'' CHR$(22).
- Module 100-290 (joystick) : La valeur renvoyée par la fonction JOY(0) est la mise en binaire à 5 chiffres (ligne 120), d'où modification des coordonnées verticale et horizontale V et H (lignes 130 et 140). Pour les détails, voir MPSA page 131.
  Les lignes 150 et 160 empêchent le curseur de sortir de la grille. Le CHR$(233) est un petit carré plein, alors que le CHR$(144) est un point centré (notre curseur). F est le témoin d'appui sur FIRE. Le bouclage du cycle provoque le clignotement de la case du curseur. Les lignes 230 et 270 sont des ralentisseurs pour mieux contrôler le joystick.
- Module Résultats (lignes 300 à 450) : On supprime d'abord le mode transparent. Pour le comptage horizontal, on utilise la fonction TEST qui donne 1 pour une case noircie. En 350, l'image binaire est reconstituée par concaténation, puis traduite en décimal en ligne 360 et mise en DIM CH.
  En 400, on vérifie s'il s'agit d'un caractère ou d'un code ASCII. Exécution du SYMBOL en 420 et affichage des résultats.
- Module Imprimante (700 à 830) : On procède de la même manière au comptage vertical. En 760, on imprime les paramètres des comptages horizontaux (SYMBOL) et verticaux, le caractère d'origine, son

code ASCII et sa nouvelle représentation (780-790).
— Final (ligne 490) : En efface le BORDER et on vide le buffer de l'imprimante, car on avait confirmé les caractères USA en ligne 60.

```
10 ' DEFCAR - REDEFINITION DE CARACTERES
  SUR ECRAN ET IMPRIMANTE - M.A.-86
20 MODE 1:BORDER 9:DEFINT A-Z
30 SYMBOL AFTER 32
40 CLS
50 LOCATE 4,10:PRINT "L'IMPRIMANTE EST-E
LLE PRETE ?(O/N)":Q$=UPPER$(INKEY$):IF Q
$="" THEN 50
60 IF Q$="O" THEN FIMP=1:PRINT #8,CHR$(2
7);CHR$(64);CHR$(27);"R";CHR$(0):GOTO 80
70 IF Q$<>"N" THEN 40
80 CLS:GOSUB 600
90 IF FIMP THEN LOCATE 2,3:PEN 3:PRINT"*
":PEN 1
100 'TRACE DE POINTS AVEC JOYSTICK
110 C$=UPPER$(INKEY$):IF C$="Q" THEN 300
120 B$=BIN$(JOY(0),5)
130 H=H+VAL(MID$(B$,2,1))-VAL(MID$(B$,3,
1))
140 V=V-VAL(MID$(B$,5,1))+VAL(MID$(B$,4,
1))
150 IF H<4 THEN H=4 ELSE IF H>11 THEN H=
11
160 IF V<3 THEN V=3 ELSE IF V>10 THEN V=
10
170 F=VAL(LEFT$(B$,1))
180 T=TEST(H*16-4,408-V*16)
190 IF T THEN PEN 0:LOCATE H,V:PRINT CHR
$(233):PEN 1
200 IF C$=" " THEN PEN 0:LOCATE H,V:PRIN
T CHR$(233):PEN 1:T=0
210 IF T=0 THEN PEN 3:LOCATE H,V:PRINT C
HR$(144):PEN 1
220 IF F=1 THEN LOCATE H,V:PRINT CHR$(23
3):T=1
230 FOR I=1 TO 150:NEXT
240 IF T THEN LOCATE H,V:PRINT CHR$(233)
250 PRINT CHR$(22);CHR$(0)
```

```
260 IF T=0 THEN LOCATE H,V:PRINT " "
270 FOR I=1 TO 100:NEXT
280 GOSUB 600
290 GOTO 110
300 ' RESULTATS
310 PRINT CHR$(22);CHR$(0);
320 'COMPTAGE HORIZONTAL
330 FOR N=1 TO 8:V=N+2:B$=""
340 FOR H=4 TO 11:B=TEST(H*16-4,408-V*16
)
350 B$=B$+RIGHT$(STR$(B),1)
360 NEXT:CH(N)=VAL("&X"+B$)
370 NEXT:IF FIMP THEN 400
380 LOCATE 2,12:PRINT "NOTEZ SON CODAGE
:":PRINT
390 FOR I=1 TO 7:PRINT CH(I);:NEXT:PRINT
 CH(8):PRINT
400 LOCATE 2,16:INPUT "QUEL CARACTERE ?
",K$:K=ASC(K$):IF VAL(K$)>31 THEN K=VAL(
K$)
410 LOCATE 25,16:PRINT "(=";K;")-->"
420 SYMBOL K,CH(1),CH(2),CH(3),CH(4),CH(
5),CH(6),CH(7),CH(8)
430 LOCATE 30,16:PRINT CHR$(K)
440 LOCATE 22,16:PRINT "( =";K;") -->  "
;CHR$(K)
450 IF FIMP THEN GOSUB 700
460 LOCATE 10,20:PRINT "AUTRE CARACTERE
?(O/N)":Q$=UPPER$(INKEY$):IF Q$="" THEN
460
470 IF Q$="O" THEN CLS:GOTO 80
480 IF Q$<>"N" THEN 460
490 BORDER 1:CLS:IF FIMP THEN PRINT #8,C
HR$(27);CHR$(64);
500 PRINT "Re-initialisez le CPC pour ef
facer les  SYMBOL qui resteraient actifs
.":PRINT
510 END
600 'GRILLE
610 PRINT CHR$(22);CHR$(1);:' MODE TRANS
PARENT
620 FOR I=240 TO 368 STEP 16
630 PLOT 48,I,2:DRAWR 128,0,2:NEXT
```

```
640 FOR I=48 TO 176 STEP 16
650 PLOT I,240,2:DRAWR 0,128:NEXT
660 LOCATE 20,5:PRINT "EFFACE=ESPACE"
670 RETURN
700 'COMPTAGE VERTICAL
710 FOR N=1 TO 8:H=N+3:B$=""
720 FOR V=3 TO 10:B=TEST(H*16-4,408-V*16
)
730 B$=B$+RIGHT$(STR$(B),1)
740 NEXT:CV(N)=VAL("&X"+B$)
750 NEXT
760 ' IMPRESSION
770 PRINT #8,"LETTRE ";K$;",CODE ASCII";
K;"--> ";
780 PRINT #8,CHR$(27);"K";CHR$(8);CHR$(0
);
790 FOR I=1 TO 8:PRINT #8,CHR$(CV(I));:N
EXT:PRINT #8
800 PRINT #8,"SYMBOL";K;:FOR I=1 TO 8:PR
INT #8,",";CH(I);:NEXT:PRINT #8,CHR$(13)
810 PRINT #8,"CODAGE VERTICAL:";:FOR I=1
 TO 8:PRINT #8,CV(I);:NEXT:PRINT #8,CHR$
(13)
820 PRINT #8,STRING$(60,"-")
830 RETURN
900 ' --------- FIN DE LISTING ---------
```

Comme vous le constatez, c'est un peu moins simple que le mode d'emploi... Mais heureusement, le Basic de l'AMSTRAD est très rapide, rapidité doublée par le DEFINT de la ligne 20.

## LA PRATIQUE DE L'IMPRESSION DES CARACTERES

La syntaxe du ''bit image'' étant lourde, il ne faudrait pas qu'elle surcharge nos listings ; il sera alors pratique de définir une chaîne BI$ qui la ''résume''. Programmez :

BI$ = CHR$(27) + ''K'' + CHR$(8) + CHR$(0)

Idem pour nos huit paramètres du caractère défini CV(1) à CV(8).

CD$ = '' '':FOR I = 1 TO 8
CD$ = CD$ + CHR$(CV(I)):NEXT

A chaque fois que nous voudrons imprimer notre caractère défini, il suffira d'écrire :

PRINT #8, BI$;CD$

Bien raccourcie l'écriture ; non ?

On complique :
Vous avez redéfini toutes les lettres minuscules de a à z (codes de 97 à 122), soit 26 CD$ que vous allez mettre en DIM CD$(26) ; lequel a été rempli par des READ sur des lignes de DATA où figurent les paramètres (codes verticaux). (Voir MPSA chapitre 12).
Le caractère à imprimer est défini par son indice, exemple CD$(3) était jadis le c minuscule.
A l'écran, vous avez tapé une chaîne A$ avec des caractères bizarres (ils ont conservé leurs codes ASCII), et vous vous voulez imprimer A$.
L'astuce est que l'indice des CD$, c'est le code ASCII moins 96...

```
410 FOR I=1 TO LEN(A$)
420 N=ASC(MID$(A$,I,1))-96
430 PRINT #8,BI$;CD$(N);:NEXT:PRINT #8
```

Et voilà le travail ! En 420, on analyse A$ caractère par caractère afin d'en extraire le code ASCII. Attention, il ne faut pas de blancs (code 32) dans A$, sinon inclure un IF qui fera imprimer normalement cet espace.
Imaginez les dizaines de lignes Basic qu'il aurait fallu taper si on n'avait pas cogité pour trouver des astuces pour simplifier.

## SYMBOL AFTER ET LA MEMOIRE

La réservation mémoire pour les SYMBOL par SYMBOL AFTER est assez gourmande. De ce fait, vous avez intérêt à utiliser des codes ASCII les plus élevés possibles, afin que le SYMBOL AFTER soit le *moins bas possible.* Par défaut, il est à 240, donc inutile de le mentionner pour les codes ASCII de 240 à 255.
Savez-vous qu'en mettant SYMBOL AFTER 256 (pas de SYMBOL possible), on récupère ainsi 128 octets ? En affichant SYMBOL AFTER 32, on se prive de $(256-32)\times 8=1\ 792$ octets... A méditer...
La zone réservée par cette commande n'aime pas qu'on la dérange, soit par un MEMORY, soit par une routine en langage machine qui viendrait empiéter sur elle. Voilà pourquoi il faut toujours programmer le SYMBOL AFTER en tout début de programme, surtout pas après avoir abaissé le HIMEM par un MEMORY.

# Chapitre VI

# LES HISTOGRAMMES

Les histogrammes sont ces représentations ''en buildings'', ces colonnes à plusieurs tranches, de différentes hauteurs, très usitées dans les revues d'économie : Ou bien chaque colonne (ou barre) représente un pays, divisé en plusieurs rubriques superposées, ou bien c'est l'inverse, une colonne par rubrique où s'empilent les quota de chaque pays.
On les appelle aussi ''diagrammes en barres''. Les plus simplistes, à éviter, sont des barres horizontales de longueurs variables. Ensuite les verticales, de simples rectangles plus ou moins hauts, très triste...
L'histogramme à la mode est en ''trois dimensions'', c'est-à-dire en perspective latérale plongeante. Traduisez par ''vu d'en haut et de côté''. Cette vision en volume et bien plus sympathique, donc adoptée. Tout logiciel présentant des histogrammes ''plats'' est tout de suite déconsidéré, ça fait bâclé. Vous savez donc ce qui vous attend, du ''3D'' et rien d'autre...
Il ne serait pas réaliste de vous proposer un logiciel spécial histogrammes dans lequel vous entreriez toutes vos données, nombres de colonnes, de rubriques, valeurs numériques à représenter, légendes, etc. En effet, c'est généralement un sous-programme dans un logiciel de calcul (statistiques, comptabilité, etc.). Donc à concevoir ''sur mesures''.
Le programme HISTO3D que nous vous proposons est plutôt une démonstration ; vous avez peu de chances de pouvoir l'utiliser tel quel dans une application pratique. Son but est de vous apprendre à en construire un, adapté à votre problème. C'est dire que nous allons expliquer comment nous avons calculé tous ses paramètres, car vous aurez à faire les mêmes calculs...
Nous vous invitons tout d'abord à taper ce très court programme, sans les REM, voire même en vous dispensant des lignes 400 à 460, mais avec les lignes 500 à 530.

```
10 ' HISTO3D-DEMO HISTOGRAMME - M.A.86
20 MODE 1:BORDER 9:DEFINT A-Z
30 DIM P(3,5):' 3 POSTES SUR 5 BARRES
40 'REMPLISSAGE DES DIM
50 DATA 100,50,40,30,205
60 DATA 100,100,39,150,47
70 DATA 100,50,80,40,38
80 DATA FRA,USA,JAP,ANG,RFA
90 DATA BEAUJOLAIS,BORDEAUX,DIVERS
100 FOR R=1 TO 3:FOR N=1 TO 5
110 READ P(R,N):NEXT:NEXT
120 FOR I=1 TO 5:READ PAYS$(I):NEXT
130 FOR I=1 TO 3:READ VIN$(I):NEXT
200 ' TRACE DES HISTOGRAMMES
210 N=0:R=0
220 FOR X=30 TO 518 STEP 122:N=N+1:' Les
 5 barres pays
230 ORY=50:' Origine Y
240 FOR R=1 TO 3:' Les 3 rubriques vins
250 FOR Y=ORY TO P(R,N)+ORY STEP 2:' Lec
ture des valeurs
260 PLOT X,Y,R:DRAWR 70,0:DRAWR 22,22:NE
XT:ORY=Y:' Trace d'une rubrique
270 NEXT
280 SX=XPOS:SY=YPOS:' Memorise curseur
290 FOR I=0 TO 22 STEP 2:PLOT SX-I,SY-I:
DRAWR -70,0:NEXT:' Dessus  des barres
300 PLOTR 33,40,1:TAG:PRINT PAYS$(N);:TA
GOFF:' Affichage pays
310 PLOT SX,SY,0:DRAWR -22,-22:DRAWR -70
,0:MOVER 70,0:DRAW SX-22,50:' Trace des
 aretes
320 NEXT
330 LOCATE 7,24:FOR R=1 TO 3:PEN R:PRINT
 VIN$(R);"  ";:NEXT:PEN 1
400 ' AFFICHAGE DES QUANTITES
420 N=0:FOR X=55 TO 543 STEP 122:N=N+1
430 TOT=0:FOR R=1 TO 3:TOT=TOT+P(R-1,N)
440 Y=50+TOT+P(R,N)/2
450 LOCATE X/16,(400-Y+8)/16:PAPER R:PEN
 0:PRINT USING"###";P(R,N)
460 NEXT:NEXT
500 ' FINAL
```

```
510 CALL &BB06
520 PAPER 0:PEN 1:BORDER 1:CLS
530 END
540 '--------- FIN DE LISTING ---------
```

## PETIT DESCRIPTIF DU PROGRAMME

Nous voulons représenter la consommation de vins français dans cinq pays : Beaujolais, Bordeaux et Divers (= 3 rubriques, ou postes). Les nombres utilisés pour cet exemple sont complètement bidon (si j'ose dire... )..
Pour chaque colonne, nous aurons de bas en haut : Beaujolais en jaune, Bordeaux en bleu ciel et Divers en rouge : le numéro de rubrique est donc le numéro de PEN.
Chaque colonne est surmontée par sa légende, le nom du pays en trois lettres. En bas de l'écran, les rubriques (vins) sont légendées dans leur couleur (nous sommes en MODE 1).
Pour donner du relief, les arêtes de nos ''buildings'' sont soulignées dans la couleur du fond (bleu foncé).
Dernier gadget : chaque bloc est légendé par sa valeur numérique (en PRINT USING). La durée totale du tracé est de 10,8 secondes. L'effet final est assez plaisant et n'a rien à envier à ce que l'on voit dans la presse.

## LE PROGRAMME

On attaque la partie explicative.

### Les valeurs numériques

Elles sont mises en DATA afin de garnir un tableau DIM P(5,3). On appelera N le numéro de pays (1 à 5) et R le numéro de rubrique (de 1 à 3). On remplit de même DIM PAY$(5) et DIM VIN$(3) (lignes 120-130), non définis puisqu'inférieurs à 10.

### La progression horizontale X

Sur l'axe des X, on dispose de 640 points. L'espace entre les colonnes (et le BORDER) sera de 30 points, soit $30 \times (5+1) = 180$. Il reste pour les colonnes $640 - 180 = 460$ points, divisés par $5 = 92$ points par colonne, que nous répartissons ainsi : 70 points pour la largeur de la ''façade'' et 22 points pour le flanc droit.
Le pas inter-colonnes est donc de $92 + 30 = 122$.
Expliquons la ligne 220 ; Notre tracé commence à $X = 30$ (colonne n° 1), ceci fait nous aurons quatre sauts à faire pour arriver à la colonne n° 5, donc une progression de $4 \times 122 = 488$. En partant de $X = 30$, nous irons jusqu'à $488 + 30 = 518$ au pas de 122. OK ?

**La progression verticale Y**
Sur l'axe des Y on dispose de 400 points. Laissons des marges de 50 en bas, 50 en haut, nos colonnes ne devront pas dépasser une hauteur de 300 points. Voilà pourquoi le total des valeurs dans nos trois rubriques ne dépasse pas 300. Dans la réalité, vous traiterez vos valeurs réelles par un facteur multiplicatif afin qu'un total ne dépasse pas 300 ''points''. Dans notre exemple, on n'a pas compliqué inutilement.
Les traits obliques qui vont dessiner les flancs de colonnes ont déjà une largeur de 22, ils monteront également de 22 points. En effet, en MODE 1, tout trait oblique qui n'est pas à 45°, apparaît un peu en marches d'escalier, et c'est moche !
ORY est l'origine Y de chaque rubrique ; au départ, elle est bien sûr à 50 (ligne 230).
L'écran vidéo fait en réalité 200 lignes et non 400 ; progresser verticalement avec un STEP de 1 ferait tracer les traits deux fois sur eux-mêmes, d'où un temps double. La progression en Y sera donc avec un STEP de deux points (ligne 250).
Dès qu'une rubrique est terminée, ORY prend la dernière valeur de Y (ligne 260) avant d'attaquer la rubrique suivante.

**Le tracé des colonnes**
Le motif tracé est un trait horizontal largeur 70 plus un trait oblique de 22,22. Avec empilage vers le haut. La couleur du trait est la valeur de R, fixée dans le PLOT X,Y,R de la ligne 260 ; inutile de le répéter pour les DRAWR qui le suivent (ça ralentirait).
A la fin de la dernière rubrique, seuls la façade et le flanc de la colonne sont tracés ; il reste à tracer le ''couvercle'', les arêtes et la légende de la colonne ; alors on mémorise les coordonnées du curseur (ligne 280) ; c'est le point en haut à ''l'arrière'' droite.

**Le tracé du dessus de colonne**
Tout est dans la ligne 290. On trace des traits horizontaux de longueur $-70$ en descendant à partir du trait supérieur oblique. Le curseur est alors dans l'angle en haut à gauche sur la ''façade''.

**La légende pays (ligne 300)**
On passe en TAG. Rappelez-vous qu'en MODE 1 un caractère en mode texte représente un carré de $16 \times 16$ points graphiques. En TAG, le curseur graphique se place en haut à gauche du carré-caractère à afficher. Il faut monter de $22 + 16 +$ marge minimale de 2 ; total $+40$ en Y. En translation horizontale d'abord $+22 +$ une marge d'au moins $10 = 32$, d'où un PLOTR 32,40,1 pour écrire en jaune.

**TRES IMPORTANT** : La chaîne à afficher en mode TAG doit *toujours* être suivie d'un *point-virgule*. Sinon des flèches coudées apparaissent à la suite de la chaîne ! Et n'oublions pas le TAGOFF.

**Le tracé des arêtes**
Il est indispensable de marquer en bleu foncé (PEN 0) les trois arêtes

''intérieures'' de la colonne (ligne 310). On replace le curseur graphique là où il avait été mémorisé (en haut à droite) ; on descend en diagonale de – 22, – 22, puis un trait horizontal vers la gauche de – 70. Voilà pour le ''couvercle''. Reste la grande arête verticale à droite de la ''façade''. On revient donc de 70 vers la droite par un MOVER et de là un DRAW (et non plus un DRAWR) vers l'angle ''avant'' droit en bas de la colonne.
On remarquera qu'à de rares exceptions près, on n'utilise que les commandes graphiques *relatives*, à savoir PLOTR, DRAWR et MOVER, bien plus souples à programmer que leurs homologues absolus PLOT, DRAW et MOVE.

**Légendes des rubriques (ligne 330)**
C'est très simple ; en ligne 24, on affiche les trois vins, chacun dans sa couleur.

**Affichage des quantités (lignes 400 à 460)**
Ce module est tout à fait optionnel car il n'est pas à l'abri de la critique. Deux raisons à cela : les chiffres ont une hauteur de 16 points, donc pour des rubriques de moins de 20 points, il y aura des chevauchements catastrophiques. Second point : sur le CPC 464, il est très scabreux de combiner TAG et la transparence (le CHR$(22)). De guerre las, l'auteur a utilisé des LOCATE sur PAPER couleur R et PEN 0. Or la position en hauteur est calculée par le programme, d'où des positionnements arrondis au plus proche, lesquels ne se placent pas au milieu des rubriques de faibles hauteurs.

**Le final (lignes 500-530)**
Le CALL &BB06 attend la frappe d'une touche, sans lui nous aurions un catastrophique ''READY'' en haut à gauche de l'écran...
Après cela, confirmation des PAPER et PEN par défaut, et CLS.

## D'AUTRES VARIANTES

Nous n'avons pas prévu de titre. Il vous sera facile de l'ajouter, mais pensez alors à abaisser le maximum des colonnes de 300 à 270, pour le placer en ligne 1, avec BORDER couleur PAPER.
Et pour plus de trois variables ? Il y a la solution de passer en MODE 0 mais c'est... envahissant. Essayons de rester en MODE 1. Pas question de tracer en PEN 4 qui se traduirait en PEN 0 ! Donc, à partir de R = 4, colorer en R + 1 ce qui vous fera repartir par le jaune (PEN 5 = PEN 1, PEN6 = PEN2, etc.). Vous pouvez différencier ces couleurs bis par un artifice : faites le tracé de ces rubriques en STEP 4 (ligne 250) ; cela donne un hachurage horizontal acceptable.

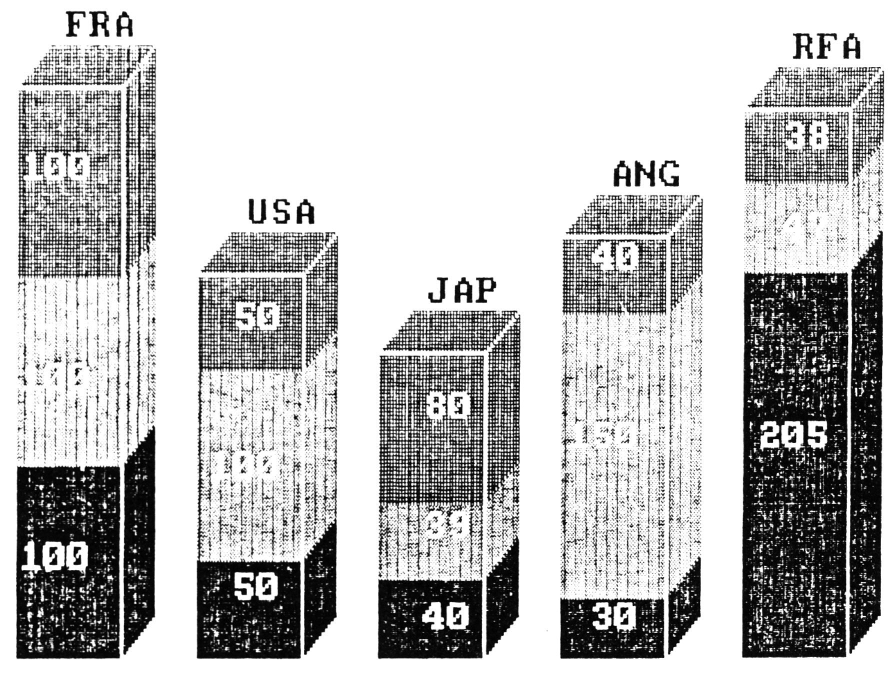
## LES PROBLEMES D'IMPRIMANTE

Bien sûr, une imprimante est monochrome (le prix des ''4 couleurs'' est encore dissuasif pour les amateurs). Nous pouvons faire une copie d'écran à l'aide d'un des programmes décrits au chapitre XVI, mais ceux-ci utilisent la fonction TEST qui renvoie la couleur du pixel à l'écran, 0, 1, 2 ou 3. L'imprimante traduit par 0 ou 1 (1 = supérieur à 0), donc nos colonnes seront toutes noires avec seulement les arêtes en blanc : pas de séparation des rubriques. Deux remèdes :
Hachurer par le STEP4 les rubriques n° 1 et 3 et le couvercle. Très simple, ajoutez une ligne 245

```
245 ST = 4:IF R = 2 THEN ST = 2
```

puis modifiez la ligne 250 : Remplacez STEP2 par STEP ST.
Enfin, ligne 290, remplacez STEP2 par STEP4. Le résultat est très présentable. Hélas, un petit défaut : nos trois noms de vins en bas de l'écran sont tous trois écrits en noir. Il faudra légender autrement.

Seconde solution : utiliser certains logiciels du commerce, tel que ''TASCOPY'' qui traduisent les différentes couleurs par des grisés différents sur imprimante.



## CONCLUSION

Comme promis, le programme était court et nous vous en avons livré tous les ''secrets''. Vous voilà donc en mesure de programmer une routine histogrammes en perspective à la suite de vos logiciels.
Une grande recommandation : lorsque vous vous attaquez à un programme graphique, ''calculez votre coup'' sur papier avec une calculatrice avant de taper une seule ligne de Basic. La tendance naturelle est de se lancer directement dans de longs tâtonnements, retouches de lignes, etc. Vous y passerez des heures pour enfin aboutir à du bancal. Il est bien plus rapide de calculer des bases solides, les programmer, et ensuite peaufiner de menus détails. La qualité est alors garantie.

# Chapitre VII

# LA REPRESENTATION CIRCULAIRE

C'est ce que l'on appelle vulgairement un ''camembert'', dont nous abreuve la télévision pour les résultats électoraux : chaque rubrique occupe un secteur de cercle plus ou moins important.
Le grand intérêt de ce type de représentation concerne les résultats exprimés en **pourcentages.** 100 % valant bien sûr 360°.
Dans un premier temps, il faut donc transformer les valeurs de chaque rubrique en pourcents par rapport au total numérique de ces rubriques.
En multipliant ces pourcents par 3,6, on obtient l'angle du secteur représentatif. OK ?
Il ne reste plus qu'à effectuer le tracé. Hélas, l'AMSTRAD ne possède pas la fonction CIRCLE, il faut donc la construire en Basic, mais pas selon la ''méthode'' du manuel d'origine que je qualifie de délirante, parce que super-lente pour conduire à un tracé en pointillés !

## LA ROUTINE CERCLE

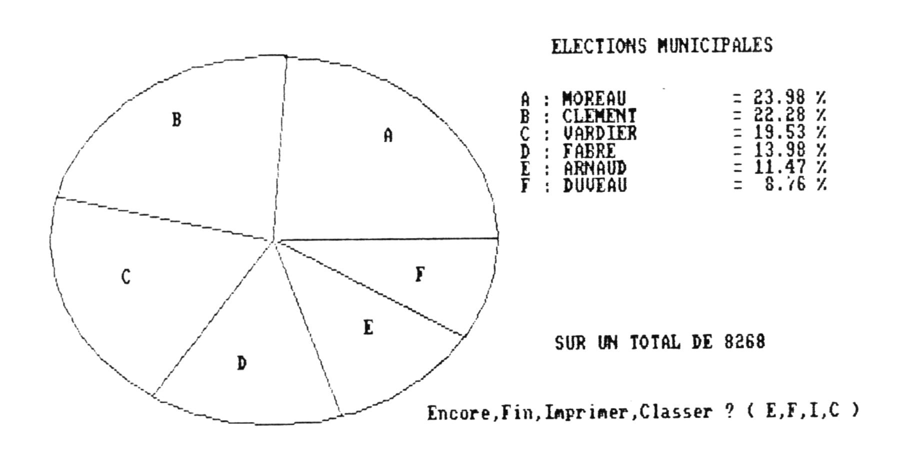
C'est un sous-programme à ajouter à la cassette ou disquette utilitaires, car appelé à être rechargé par MERGE, d'où ses numéros de lignes en 54000.
Ces trois lignes Basic tracent un cercle parfait en 1,5 seconde, et pas en pointillés, en MODE 0, 1 et 2. Le principe consiste tout simplement à tracer 36 ''cordes'' et ce polygone à 36 côtés est assimilable à un cercle, même en MODE 2.
Il suffit de donner les coordonnées X,Y du centre, le rayon et la couleur COL du PEN. Cette durée de 1,5 seconde ne varie pas avec le rayon ou le MODE.

```
54000 '  TRACE DE CERCLE
54010 DEG:PLOT X+R,Y,COL
54020 FOR A%=0 TO 360 STEP 10
54030 DRAW R*COS(A%)+X,R*SIN(A%)+Y:NEXT
54040 RETURN
```

```
            ELECTIONS MUNICIPALES

1 : MOREAU           = 23.98 % ( 1983 )
2 : CLEMENT          = 22.28 % ( 1842 )
3 : VARDIER          = 19.53 % ( 1615 )
4 : FABRE            = 13.98 % ( 1156 )
5 : ARNAUD           = 11.47 % ( 948 )
6 : DUVEAU           =  8.76 % ( 724 )

             Sur un TOTAL de 8268
```



Essayez :
1 CLS:X=320:Y=200:R=190:COL=1
2 GOSUB 54000:END

Ceux d'entre vous qui ne connaissent pas (ou ont oublié) les fonctions trigonométriques COS et SIN pourront se reporter à MPSA pages 56-58. Disons brutalement que COS varie dans le sens X et SIN dans le sens Y. Comme COS et SIN ne peuvent dépasser la valeur 1, ils sont multipliés par le rayon, exprimé en ''points'' graphiques AMSTRAD.

## LE PROGRAMME CAMEMBERT

Un écran de saisie vous invite à entrer les ''composantes'', c'est-à-dire chaque nom de rubrique puis sa valeur numérique. Par exemple, pour une élection, nom du candidat, ENTER, nombre de voix, ENTER. En cas d'erreur, tapez ''E'' pour revenir un ''cran'' en arrière.
Le nombre de ''rubriques'' doit être compris entre 3 et 12.
En fin de saisie, tapez Q pour quitter. L'écran vous propose un tri (classement) puis c'est la représentation graphique en haute résolution (MODE 2) :
Les divers secteurs sont légendés en leur centre par une lettre (A, B, C...). A droite apparaît la légende lettre-nom-pourcentage. En-dessous, le total numérique (exemple nombre de suffrages exprimées) ; et, plus bas, quatre options : E=Encore (c'est un RUN), C=Classement (ordre croissant), I=Imprimer la liste complète des résultats, F=FIN (END+CLS en MODE 1). Dans le module impression, vous pourrez enchaîner sur une copie d'écran par un de nos programmes du chapitre XVI.
Nous avons ajouté un petit gadget, un ''LOGO'' de titre, qui écrit ''représentation camembert'' en deux couleurs, avec toutes ces lettres réparties en cercle (lignes 2000 à 2140). On vous demande d'entrer le nom de ce tableau ; titre qui sera centré sur l'écran final.
On peut donc dire que ce programme se divise en trois parties :
1 — Saisie et calcul des données,
2 — Représentation graphique des données,
3 — Des modules facultatifs : logo, imprimante, tri.

```
10 ' CAMEMBERT / AMSTRAD CPC / M.A. 86
20 DIM NM$(12),CA(12,3)
30 'DIM CA:1=QUANTITE NB;2=NB/TOTAL;3=PO
SITION ANGULAIRE
40 INK 0,1:INK 1,24:INK 2,20:INK 3,6:BOR
DER 15
50 'TITRE
60 GOSUB 2000:' LOGO
```

```
70 LOCATE 3,20:PEN 2:PRINT "Nom du Table
au: ";STRING$(22,".");STRING$(18,"."):PE
N 1:LOCATE 19,20:LINE INPUT TIT$
80 TIT$=LEFT$(TIT$,40)
100 'SAISIE
110 BORDER 2:PAPER 2:PEN 3:CLS:LOCATE 3,
2:PRINT "12 NOMS maxi de 15 CARACTERES m
axi.
120 LOCATE 4,24:PRINT "Erreur = E ; pour
 Quitter NOM = Q"
130 LOCATE 10,4:PRINT "NOM:":LOCATE 25,4
:PRINT "QUANTITE:":PEN 0
140 N=N+1:LOCATE 5,5+N:PRINT USING "##";
N:LOCATE 7,5+N:PRINT " : ";STRING$(15,".
"):LOCATE 10,5+N:LINE INPUT "",NM$
150 IF UPPER$(NM$)="Q" THEN N=N-1:GOTO 3
00
160 IF UPPER$(NM$)="E" THEN N=N-2:TNB=TN
B-NB:GOTO 140
170 NM$(N)=LEFT$(NM$,15)
180 LOCATE 27,5+N:INPUT "",NB$:NB=VAL(NB
$):IF UPPER$(NB$)="E" THEN N=N-1:GOTO 14
0
190 IF (NB=0 AND NB$<>"0") OR NB<0 THEN
PRINT CHR$(7):GOTO 180
200 CA(N,1)=NB:TNB=TNB+NB:IF N=12 THEN 3
00
210 GOTO 140
300 'FIN DE SAISIE
310 PEN 3:IF N=1 THEN PRINT CHR$(7):LOCA
TE 3,22:PRINT "Pas de Camembert pour UNE
 variable...":FOR I=1 TO 3000:NEXT:RUN
320 LOCATE 6,22:PRINT "Voulez-vous class
er ? ( O/N )"
330 Q$=UPPER$(INKEY$):IF Q$="N" THEN 500
340 IF Q$<>"O" THEN 330
350 LOCATE 6,22:PRINT CHR$(18);SPC(11);"
Patience..."
400 'TRI
410 FOR J=1 TO N:E=CA(J,1)
420 FOR I=J TO N
430 IF CA(I,1)>=E THEN E=CA(I,1):K=I:E$=
NM$(I)
```

```
440 NEXT
450 CA(K,1)=CA(J,1):CA(J,1)=E
460 NM$(K)=NM$(J):NM$(J)=E$
470 NEXT
500 'CALCULS DIM CA
510 FOR I=1 TO N:CA(I,2)=CA(I,1)/TNB:CA(
I,3)=360*CA(I,2)+CA(I-1,3):NEXT
600 ' TRACE
610 MODE 2:INK 0,13:INK 1,0:BORDER 9
620 X=180:Y=200:R=170:COL=1:GOSUB 54000
630 FOR I=1 TO N:PLOT X,Y,1
640 DRAWR COS(CA(I,3))*R,SIN(CA(I,3))*R:
NEXT
650 FOR I=N TO 1 STEP-1
660 AB=(CA(I,3)+CA(I-1,3))/2:'ANGLE BISS
ECTRICE
670 PLOT X,Y
680 DRAWR COS(AB)*120,SIN(AB)*120,0
690 LOCATE XPOS/8,(400-YPOS)/16:PRINT CH
R$(64+I):NEXT:PLOT X,Y,1
700 LOCATE 40+(40-LEN(TIT$))/2,2:PRINT T
IT$
710 FOR I=1 TO N
720 LOCATE 47,4+I:PRINT CHR$(64+I);" : "
;NM$(I)
730 LOCATE 67,4+I:PRINT "= ";:PRINT USIN
G "##.##";CA(I,2)*100;:PRINT " %"
740 NEXT
750 LOCATE 50,19:PRINT "SUR UN TOTAL DE"
;TNB
760 LOCATE 38,23:PRINT "Encore,Fin,Classe
er,Imprimer ? ( E,F,C,I )"
770 Q$=UPPER$(INKEY$):IF Q$="E" THEN RUN
780 IF Q$="C" THEN 400
790 IF Q$="I" THEN GOSUB 3000:GOTO 770
800 IF Q$<>"F" THEN 770
810 MODE 1:INK 0,1:INK 1,24:BORDER 1:PAP
ER 0:PEN 1:CLS
820 LOCATE 11,12:PRINT "VOUS POUVEZ ETEI
NDRE.":PRINT:PRINT
830 END
2000 ' LOGO
2010 MODE 1:DEG:R=120
```

```
2020 ORIGIN 320,260:TAG
2030 A$="REPRESENTATION"
2040 B$="CAMEMBERT"
2050 FOR A=172 TO 0 STEP-(180/LEN(A$)):I
=I+1
2060 PLOT COS(A)*R,SIN(A)*R,1
2070 PRINT MID$(A$,I,1);
2080 NEXT
2090 FOR A=207 TO 340 STEP (140/LEN(B$))
:J=J+1
2100 PLOT COS(A)*R,SIN(A)*R,3
2110 PRINT MID$(B$,J,1);
2120 NEXT
2130 TAGOFF:ORIGIN 0,0:PEN 1
2140 RETURN
3000 'IMPRESSION (EPSON)
3010 PRINT#8,CHR$(27)"@";SPC((40-LEN(TIT
$))/2);TIT$:PRINT#8
3020 FOR I=1 TO N
3030 PRINT#8,USING "##";I;:PRINT#8," : "
;
3040 PRINT#8,NM$(I);SPC(15-LEN(NM$(I)));
" = ";
3050 PRINT#8,USING"##.##";CA(I,2)*100;
3060 PRINT#8," % (";CA(I,1);")"
3070 NEXT
3080 PRINT#8:PRINT#8,SPC(12);"Sur un TOT
AL de";TNB
3090 FOR I=1 TO 4:PRINT#8:NEXT
3100 RETURN
54000 '  TRACE DE CERCLE
54010 DEG:PLOT X+R,Y,COL
54020 FOR A%=0 TO 360 STEP 10
54030 DRAW R*COS(A%)+X,R*SIN(A%)+Y:NEXT
54040 RETURN
65535 ' ------- FIN DE LISTING -------
```

```
VARIABLES DE CAMEMBERT


- A , A% : ANGLE EN DEGRES
- A$,B$ : TEXTE DU LOGO
- AB : ANGLE DE LA BISSECTRICE
- CA(12,3) : RESULTATS CALCULES
- COL : COULEUR CERCLE
- E,E$ : VALEURS BULLES POUR TRI
- I,J,K : INDICES DE COMPTAGES
- N : NUMERO RUBRIQUE
- NB$ , NB : NOMBRE ENTRE
- NM$(12) : NOMS DES RUBRIQUES
- Q$ : REPONSE A QUESTION
- R : RAYON
- TIT$ : TITRE
- TNB : TOTAL NOMBRES ENTRES
- X,Y : COORDONNEES DU CENTRE
```

## SAISIE ET CALCUL DES DONNEES

Les valeurs numériques sont logées dans le DIM CA(12,3). Les trois indices ''horizontaux'' sont ''1'' la valeur NB saisie à l'écran, ''2''' ce nombre NB divisé par le total TNB, ''3'' la position en degrés dans le cercle de la ''rubrique'' en question. Les valeurs ''2'' seront bien sûr calculées après la saisie, les valeurs ''3'' seront recalculées en cas de tri.
Si en cours de saisie on tape E (erreur) en guise de nom, le curseur invite à entrer le nom puis la valeur de la rubrique précédente. Si E est tapé en guide de valeur, le curseur invite à ré-entrer le nom de la rubrique en cours, puis la valeur.
La fin de saisie, obtenue en tapant Q en guise de nom ou après la 12e rubrique, propose le tri par valeurs croissantes.
C'est la ligne 510 qui va remplir les indices 2 et 3 du tableau DIM CA (12,3).

## LE TRACE (lignes 600 à 690)

On passe en MODE 2, traits noirs sur fond gris. On trace le cercle dans la moitié gauche puis les rayons séparateurs de rubriques (630,640). Toujours à l'aide des CA(N,3), on calcule l'angle de la bissectrice de chaque secteur (660). On trace en PEN 0 (invisible) des rayons plus courts sur ces bissectrices (680) et, en bout de ces rayons, on affiche une lettre A, puis B, etc. (690), lettre qui a légendé le secteur.

## L'AFFICHAGE (lignes 700 à 800)

Le titre se centre dans la moitié droite en haut de l'écran (700). Puis, toujours dans la moitié droite, les légendes, à savoir la lettre représentative, le nom de la rubrique et le résultat en pourcents. En final, le total TNB.
Enfin un menu d'interventions : Encore, Tri, Imprimer et Fin.
L'option Imprimer se contente d'écrire les résultats complets. Libre à vous de la continuer par un de nos sous-programmes de Hard-copy d'écran.

**NOTE** : Pour positionner des lettres sur le curseur graphique vous remarquerez que l'on peut tout aussi bien utiliser TAG, comme dans le logo du module 2000, ou tout simplement LOCATE positionné par XPOS et YPOS (ligne 690).

## CONCLUSION

Notre camembert est très net (MODE 2) mais en monochrome ; son tracé est très rapide. Sans chauvinisme, nous le préférons aux camemberts avec secteurs colorés ''tape-à-l'œil'' mais très difficilement lisibles. Comme pour nos histogrammes du chapitre précédent, on peut prétendre à la ''qualité pro''. Il va de soi que vous pourrez adapter ce programme à la suite d'un de vos logiciels de calcul, ce qui supprime bien sûr toute la partie saisie.
Notre limitation à douze rubriques-secteurs est tout à fait arbitraire,, mais au-delà ce serait plutôt une roue de bicyclette, et la représentation perdrait tout impact visuel parce que trop surchargée.

# Chapitre VIII

# LES DIAGRAMMES

Il s'agit de graphiques classiques en ''X,Y''. La difficulté rebutante est sans conteste la graduation des axes, il fallait rendre cela automatique. Le logiciel Basic que nous vous proposons se charge de tout ; vous entrez seulement les mini et maxi en X et Y, les noms des unités et le titre du graphe, il décide alors du pas des graduations, des marquages de valeurs, il trace les axes, centre le titre et légende les axes. Quant aux valeurs numériques à représenter, deux méthodes d'entrées : un tableau de saisie où les valeurs de chaque point sont entrées au clavier. Ou vous entrez une formule mathématique même fort longue et complexe (maxi 255 caractères...). Le logiciel calcule alors 101 points pour en tracer la courbe représentative.
A tout cela s'ajoutent quelques options très utiles, enregistrement d'un graphique, rechargement, édition sur imprimante des valeurs X, Y des points (saisis ou calculés) et hard-copy de l'image.
Contrairement au programme précédent sur les histogrammes, ce logiciel est ''complet'', indépendant, une ''bête à tracer''. Est-il vraiment universel ? Il ne lui manque qu'une option, que nous n'avons pas voulu ajouter car elle aurait ''effrayé'' la plupart d'entre vous, il s'agit des échelles logarithmiques. Si vous avez besoin de ces progressions, vous êtes forcément capable d'écrire ce supplément (environ 1500 octets en plus). C'est en fait beaucoup plus simple que pour nos échelles ''décimales''.

## LE PRINCIPE DU PROGRAMME

L'utilisateur qui souhaite virevolter dans ce logiciel doit seulement bien avoir en esprit le fait suivant :
Le tableau des données à représenter sous forme de courbe est *totalement indépendant* de l'ensemble des paramètres du tracé (limites d'axes, titre, légendes, etc.).

Avantages : On peut conserver la présentation du graphe pour d'autres données numériques, ou garder ces dernières pour une autre présentation. La sauvegarde concerne les paramètres suivis des données. Après un rechargement, on peut modifier l'un ou l'autre des deux ensembles. Donc souplesse avec bidouilles sans limitation (pas de SAVE de l'écran).
C'est le sous-programme ''Tracé'' qui assemble ces deux domaines ; il peut enchaîner sur un hard-copy d'écran.
D'autre part, tout le logiciel est en programmation ''structurée'', c'est-à-dire une suite de modules indépendantes appelés par des GOSUB à partir du menu. Il vous sera alors très facile de le personnaliser par la suite ; suppression, ajout, modification, récupération de certains modules pour d'autres programmes, etc. Bref, tous les atouts d'un logiciel structuré.
En cas d'erreur lors de l'utilisation, il y a une issue de secours : touche ESC, puis tapez GOTO 1000 (facile à retenir), d'où retour au menu sans rien perdre.

## LE LISTING

Certes, il est long mais vous pouvez vous dispenser de taper certains modules (instructions, impression). Le ''très gros morceau'' est bien sûr le tracé des axes gradués ; c'est très complexe avec beaucoup de IF, de flags, de variables calculées, etc. Ce fut un grand casse-tête pour prévoir tous les cas possibles. Comme la définition d'écran est de 200 points en Y contre 640 points en X, il y a deux sous-programmes différents.
Les autres modules sont en revanche simples à comprendre et à écrire.

**TRES IMPORTANT** : Ligne 130, nous chargeons un programme en langage machine très court (181 octets) décrit au chapitre XVI, que vous pourrez taper avec le ''MACHDIM'' du chapitre IV. Si cela vous rebute encore, ne tapez pas cette ligne 130. En contre-partie, pour avoir la copie d'écran sur imprimante, vous ajouterez alors en fin de listing la routine Basic ''Hardtest'' (chapitre XVI), qui fait exactement la même chose mais en beaucoup plus lent ; et ligne 11420 vous remplacerez CALL &9F00 par GOSUB 58000. C'est tout. Evidemment, si vous n'avez pas d'imprimante, ce sera encore plus simple...

## LA DEFINITION DES AXES
## (Option A - lignes 2000 à 2600)

On vous demande d'entrer les ''bornes'' mini et maxi pour chacun des axes X (horizontal) et Y (vertical) ; là, deux obligations (sinon refus) : il faut que ces nombres **ainsi que leurs différences** (dX et dY) soient des nombres entiers (=non décimaux) et ayant au maximum deux

''chiffres significatifs '' (=suivis de zéros). Des exemples seront plus clairs :

- mini=50, maxi=160 (d=110) : accepté
- mini=950, maxi=2200 (d=1250) : refusé,

parce que 1250 a trois chiffres significatifs (1, 2 et 5). On corrigera donc par mini=900.
Par la même occasion, on vous demande (mais c'est facultatif) le titre du graphe, les unités en X et les unités en Y. Puis le programme calcule (très rapidement !) toute une série de coefficients ; voir la liste des variables. Je vous fais grâce des détails de ces calculs car ce serait long et imbuvable pour beaucoup d'entre vous.

## LE TRACE DES AXES
## (Lignes 3000 à 4200)

L'écran passe en MODE 2, traits noirs sur fond gris. Après le tracé des deux axes, c'est le tour des petits traits de graduations (3 hauteurs différentes) au pas de 1, 2 ou 5, décidé par le programme. Ensuite, ce sont leurs valeurs, bien centrées en face du trait.
Pour la clarté de la lecture, il n'écrit que les deux premiers chiffres, sauf le nombre ''100'' et quand il est en bout d'échelle. Ainsi, une échelle de 28000 à 87000 sera en fait légendée de 28 à 87, mais les ''butées'' sont rappelées à l'écran ; toutefois, une échelle de −4 à +9 sera effectivement graduée de −4 à +9.
Cas d'une échelle 0 à 100 (ou 0 à 1000... ). En X on va de 0 à 100 au pas de 1 avec indications des valeurs tous les dix. Trois hauteurs de traits pour unités, valeurs en 5 (5, 15, 25,...) et les valeurs en 10 (10, 20, 30,...) ; donc exactement comme sur un double décimètre. Sur l'axe Y, marquage tous les dix mais les graduations sont ici au pas de 2, car verticalement on n'a que 200 points *réels* au lieu de 640 horizontalement.
Cette finesse des graduations d'axes est le point fort de ce logiciel, car inhabituelle, même sur les logiciels graphiques professionnels (sur IBM-PC, etc...).
Viennent ensuite les inscriptions (légendes). Le titre est auto-centré en haut. A mi-longueur d'axe, la légende de l'axe, et en bout d'axe, l'unité utilisée pour l'échelle. En début d'axe, le rappel des mini et maxi réels de l'échelle. A noter que pour l'axe Y, la légende s'inscrit *verticalement*. Bref, on n'a pas lésiné sur la présentation ; standing oblige...

## L'ENTREE DES DONNEES NUMERIQUES
## (Lignes 5000-5400)

L'option ''D'' conduit à un tableau de saisie des valeurs X et Y ; à gauche apparaît le numéro d'entrée de chaque point. Ces valeurs sont stockées dans un DIM PT(102,1). Pour quitter la saisie, tapez la lettre ''Q''. Ces couples de valeurs sont alors triées par valeur x croissante, ce en vue du tracé.

## LE TRACE DE LA COURBE
## (Lignes 8000 à 8200)

C'est un module court puisqu'il reprend le tracé des axes gradués, puis c'est une suite de DRAW de "point" en "point", en prenant les coordonnées dans le tableau DIM PT.
En fin de dessin, on peut revenir au menu en frappant une touche quelconque.

## IMPRESSION
## (Lignes 11000 à 11620)

Un sous-menu propose de lister les valeurs du tableau DIM (même avec la DMP1) ou le hard-copy de l'écran. Cette dernière option provoque le tracé à l'écran. En fin d'impression, retour automatique au menu.

## SAUVEGARDE ET CHARGEMENT
## (Lignes 9000 à 10500)

La syntaxe est aussi compatible avec les cassettes.
Le programme ajoute un nom d'extension ".GRA", très utile pour identifier *l'origine* d'un fichier sur un catalogue de disquette.
Pour rappeler ce fichier, il n'est pas nécessaire de faire suivre le nom par ".GRA".
A la sauvegarde, si vous répondez par ENTER à la question nom, ce dernier sera constitué par les huit premiers caractères du titre TIT$.
Vous remarquerez qu'en lignes 9070 et 9080 tous les paramètres sont sauvegardés. De ce fait, après un chargement, il suffit de demander l'option T (tracé).

## L'ENTREE DES FORMULES
## (Lignes 6000 à 7100)

Cette option "F" étant un peu plus complexe, nous l'avons gardée pour la fin. Disons aux "matheux" qu'elle permet aussi les coordonnées polaires, par exemple dessiner un cercle, spirale, etc. mais en deux passes.
Le mode d'entrée d'une formule fait appel à une particularité des AMSTRAD CPC, le fait que l'on peut modifier un programme Basic arrêté par END et relancer par GOTO sans pour cela perdre les variables en cours (le cas est pratiquement unique, n'espérez donc pas réutiliser cette technique sur un autre micro-ordinateur).
Donc, à l'appel de cette fonction, le programme s'arrête et vous devez taper la ligne 7000 contenant votre formule ; par exemple tapez :

7000 Y=10/X puis ENTER

Ceci fait, pressez la touche "." du pavé numérique ; cela provoque le redémarrage par un GOTO 6500.

**TRES IMPORTANT** : Avant de lancer l'option F, il est impératif d'avoir déjà défini les axes (option A), sinon refus. Supposons que pour X et Y nous ayons fixé mini = 1 et maxi = 10.
Le programme va alors appliquer 102 fois cette formule avec X variant de X mini à X maxi, d'où 102 valeurs de Y ; le tout est mis en DIM PT (lignes 6510 à 6590). Les valeurs mini et maxi de Y ainsi calculées sont mémorisées et affichées en fin de calculs. Si elles sont différentes de celles prévues lors de l'option A vous avez alors la possibilité de les corriger.
Faisons un essai :

a — Option A. X mini = 1 ; X maxi = 10 ; Y mini = 1 ; Y maxi = 10. Répondons par ENTER aux questions titre, légendes, unités. Retour au menu.

b — Option F. Tapez 7000 Y = 10/X, Enter, point décimal du pavé. Un message nous annonce que les calculs de Y ont donné comme limites 0.991080279 et 10. On ne modifie pas. Retour au menu.

c — Option T. On obtient alors une superbe hyperbole en haute résolution, sur des axes gradués de 1 à 10.

d — Une touche quelconque pour revenir au menu, où l'on redemande l'option A : pour :X et Y fixons mini = 0, maxi = 12. Menu.

e — Option T. Notre hyperbole ne vient plus toucher les axes.

f — Revenez en option A, mêmes limites mais Légende X = ''RESISTANCE'' ; Unité X = ''K.OHM''; Légende Y = ''INTENSITE'' ; Unité Y = ''mA''; Titre = ''LOI D'OHM sous 10 VOLTS''. Menu. Option T.

Tout ceci pour vous prouver l'indépendance des données numériques vis-à-vis des paramètres de l'option A.

## LE TRACE EN DEUX PASSES

Nous abordons le ''second degré'' ; problème, représenter un cercle. Examinez les lignes 6530 et 6550 : vous remarquerez que ce qui est mis en DIM n'est pas X mais Z = X, à moins que Z soit calculé en fonction de X, toujours dans la ligne 7000.

a — Option A : X mini = 0 ; X maxi = 360. (C'est l'angle en degrés). Y mini = – 150 ; Y maxi = 150 (c'est le sinus de l'angle multiplié arbitrairement par 150 points).

b — Option F ; tapez :
7000 DEG:Y = SIN(X) * 150:Z = COS(X) * 150

c — Option A. En effet, c'est Z que l'on veut représenter sur l'axe horizontal. Essayons pour X et Y mini = – 170 et maxi = 170.

d — Option T. C'est plutôt une élipse ! Il faut comprimer l'échelle X.

e — Option A, encore : X mini = – 250 ; X maxi = 250 ; Y mini = – 170 ; Y maxi = 170.

f — Option T : ça, c'est un beau cercle !

Encore plus fort ! Traçons une spirale faisant deux tours, donc angle variant de 0 à 720 degrés :

a — Option A : X de 0 à 720 : Y de 0 à 10 : On se moque de ces limites de Y puisque nous devons refaire l'option A...

b — Option F. Entrez la ligne formule.
7000 DEG:Y = (1 + X/720)*SIN(X):Z = (1 + X/720)*COS(X)
Pas de modification des limites Y, d'où menu.

c — Option A : X de – 3 à 3 ; Y de – 2 à 2 ; TITRE = la formule.

d — Option T. Une superbe spirale à deux tours.

Les passionnés de mathématiques vont pouvoir s'en donner à cœur joie puisque l'AMSTRAD trace en cinq secondes ce qui aurait demandé 30 minutes de développement avec une calculatrice...
Pour des calculs super complexes, vous pouvez taper les lignes 7010 à 7090, mais n'oubliez pas de les effacer pour la formule suivante avant de lancer les calculs par le point décimal.

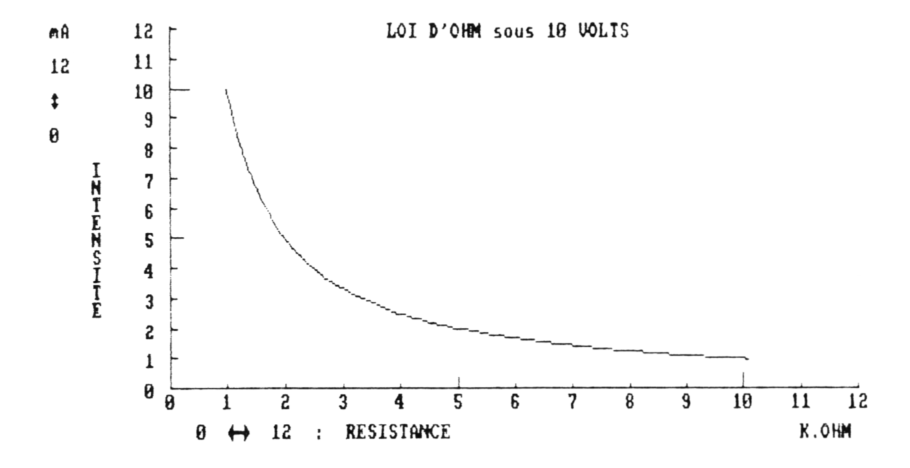
**Usage de l'imprimante**
Notre copie d'écran provoque un ''étirement'' horizontal. De ce fait, un cercle à l'écran devient une élipse sur papier. Pour remédier à cela, en option A, augmentez d'un tiers les limites de X.
Au menu, quittez par l'option ''Q''. Vous pouvez alors imprimer votre formule par :

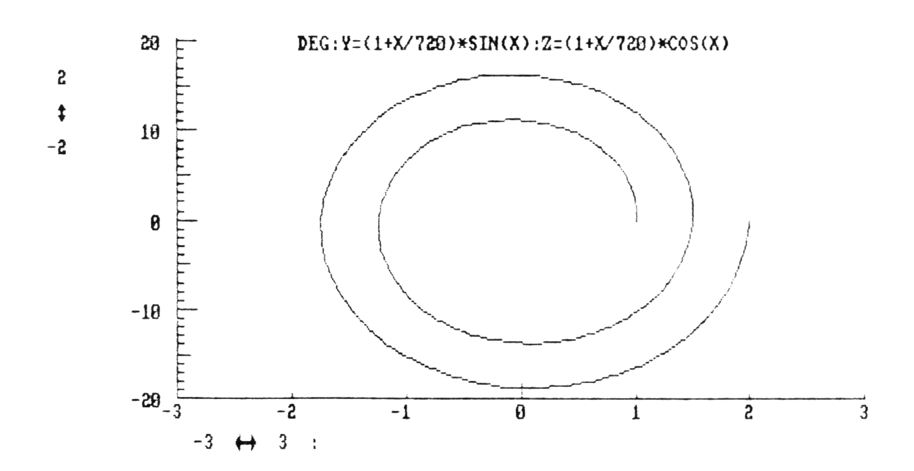
LIST 7000, #8

**NOTE** : Le programme est truffé de sécurités anti-plantages. Exemple, le module des lignes 20000 ; en cas de division par zéro ou d'overflow, la valeur de Y prend la valeur calculée précédente. Lors du tracé, cela se traduira par un écrêtage de la courbe.

## CONCLUSION

Ce programme utilitaire fait moins de 9000 octets plus un peu plus de


LOI D'OHM sous 10 VOLTS
mA
12
↕
0
INTENSITE
12
11
10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12
0 ↔ 12 : RESISTANCE
K.OHM


DEG:Y=(1+X/720)*SIN(X):Z=(1+X/720)*COS(X)
2
↕
-2
20
10
0
-10
-20
-3 -2 -1 0 1 2 3
-3 ↔ 3 :

1000 octets pour le tableau DIM, c'est dire que vous avez largement la place pour le grossir par des options ou des ''conditionnements'' propres à vos problèmes graphiques personnels.
Si vous désirez sauvegarder aussi l'image de l'écran, programmez :

```
EXT = INSTR(FICH$,''.''):ECRAN$ = FICH$
MID$(ECRAN$,EXT,4) = ''.SCR''
SAVE ECRAN$,B,&C000,&4000
```

Le début de ces instructions consiste à remplacer l'extension ''.GRA'' par ''.SCR'' (de screen).

```
VARIABLES DE GRAPHMAT:


- C$ : Chiffre droite de Graduation
- D$ : Donnee entree
- DX,DY : DELTA X,Y
- DXE,DYE : DELTA echelles X,Y
- FAXY : Flag Axes X,Y definis
- FICH$ : Nom du Fichier enregistre
- FIMP : Flag demande d'Impression
- FQ : Flag demande de quitter
- FTRA : Flag demande de Trace
- H : Hauteur d'un Repere
- I,K,N : Indices de Comptages
- LGX$,LGY$ : Legendes X,Y
- MINY,MAXY : MIN,MAX des Y calcules
- MX,MY : Maxi X,Y
- MXE,MYE : Maxi echelle X,Y
- NF : Nombre total de Points
- OX,OY : Origines X,Y
- OXE,OYE : Origines Echelle X,Y
- PT(102,1) : Coordonnees des Points
- RGX,RGY : Rapports graphiques
- RGXE,RGYE : Rapp. graphique d'Echelle
- ST,STX : STEP de FOR NEXT
- TEX$ : Liste d'Options
- TIT$ : Titre du Graphique
- UX$,UY$ : Unites X,Y
- VG : Valeur d'une Graduation
- Z : Valeur de X dans PT(102,1)
```

```
10 ' GRAPHMAT - TRACEUR DE GRAPHIQUES -
 ( + HARD9F00 )
20 'AMSTRAD CPC - M.ARCHAMBAULT 3/1986
30 OPENOUT"BIDON":MEMORY HIMEM-1:CLOSEOU
T
40 ON ERROR GOTO 20000
100 DIM PT(102,1)
110 DEFINT I,J,K,N:ORIGIN 0,0
120 KEY 138,"GOTO 6500"+CHR$(13)
130 LOAD "HARD9F00":' Programme de Hard
Copy en Langage Machine (facultatif).
200 'TITRE
210 MODE 1:PAPER 1:PEN 3:CLS
220 LOCATE 13,5:PRINT "G R A P H M A T":
PEN 2
230 LOCATE 4,10:PRINT "Michel Archambaul
t - AMSTRAD CPC":PEN 0
235 PLOT 0,0,3:DRAW 150,380:DRAW 300,100
:DRAW 500,200:DRAW 640,0
240 LOCATE 10,15:PRINT "Instructions , M
enu ?":TEX$="IM":GOSUB 50000
250 PAPER 0:PEN 1:IF K=1 THEN GOSUB 1500
0
1000 'MENU
1010 INK 0,1:INK 1,24:BORDER 1
1020 MODE 1:CLS
1030 PEN 3:LOCATE 2,2:PRINT "T R A C E U
 R  DE  G R A P H I Q U E S":PEN 1
1040 LOCATE 10,5:PRINT "A - DEFINITION D
ES AXES"
1045 LOCATE 10,7:PRINT "V - VISION DES A
XES"
1050 LOCATE 10,9:PRINT "D - ENTREES DES
DONNEES"
1060 LOCATE 10,11:PRINT "F - ENTREE D'UN
E FORMULE"
1070 LOCATE 10,13:PRINT "T - TRACE DU GR
APHIQUE"
1080 LOCATE 10,15:PRINT "S - SAUVEGARDE"
1090 LOCATE 10,17:PRINT "C - CHARGEMENT"
1095 LOCATE 10,19:PRINT "I - IMPRIMER"
1100 LOCATE 10,21:PRINT "Q - QUITTER"
```

```
1110 PEN 2:TEX$="AVDFTSCIQ":GOSUB 50000:
PEN 1
1120 ON K GOSUB 2000,3000,5000,6000,8000
,9000,10000,11000,12000
1130 GOTO 1000
2000 'ENTREE DES CARACTERISTIQUES
2010 CLS:FTRA=0
2020 PEN 3:LOCATE 1,1:PRINT "Nombres ent
iers/2 chiffres significatifs":PEN 1
2030 PEN 2:LOCATE 10,3:PRINT "AXE X ( ho
rizontal ):":PEN 1
2040 LOCATE 10,5:INPUT"Limite X MINI: ",
OX$:OX=VAL(OX$)
2050 LOCATE 10,6:INPUT"Limite X MAXI: ",
MX$:MX=VAL(MX$)
2060 DX=MX-OX:IF DX<=0 THEN PRINT CHR$(7
):GOTO 2040
2070 IF VAL(MID$(STR$(DX),4,1)) >0 THEN
PRINT CHR$(7);"dX a 3 chiffres significa
tifs:";DX:FOR I=1 TO 6000:NEXT:GOTO 2000
2080 INPUT"  LEGENDE X: ",LGX$
2090 INPUT"  UNITES X: ",UX$
2100 PEN 2:LOCATE 10,10:PRINT "AXE Y ( v
ertical ):":PEN 1
2110 LOCATE 10,12:INPUT"Limite Y MINI: "
,OY$:OY=VAL(OY$)
2120 LOCATE 10,13:INPUT"Limite Y MAXI: "
,MY$:MY=VAL(MY$)
2130 DY=MY-OY:IF DY<=0 THEN PRINT CHR$(7
):GOTO 2100
2140 IF VAL(MID$(STR$(DY),4,1)) >0 THEN
PRINT CHR$(7);"dY a 3 chiffres significa
tifs:";DY;:FOR I=1 TO 6000:NEXT:PRINT CH
R$(17):GOTO 2100
2150 INPUT"  LEGENDE Y: ",LGY$
2160 INPUT"  UNITES Y: ",UY$:PRINT
2170 INPUT"  TITRE DU GRAPHIQUE: ",TIT$
2200 'COEFFICIENTS X
2210 RGX=520/DX
2220 DXE=VAL(LEFT$(STR$(DX),3))
2230 IF DXE=10 AND DX<>10 AND OX=0 THEN
DXE=100 ELSE IF DXE<10 AND DX<5 THEN DXE
=DXE*10
```

```
2240 RX=DX/DXE
2245 IF MX/RX>100 THEN DXE=DXE/10:GOTO 2
240
2250 RGXE=520/DXE
2260 OXE=ROUND(OX/RX):MXE=ROUND(MX/RX)
2300 'COEFFICIENTS Y
2310 RGY=330/DY
2320 DYE=VAL(LEFT$(STR$(DY),3))
2330 IF DYE=10 AND DY<>10 AND OY=0 THEN
DYE=100 ELSE IF DYE<10 AND DY<5 THEN DYE
=DYE*10
2340 RY=DY/DYE
2345 IF MY/RY>100 THEN DYE=DYE/10:GOTO 2
340
2350 RGYE=330/DYE
2360 OYE=ROUND(OY/RY):MYE=ROUND(MY/RY)
2500 PRINT:PRINT TAB(6);"Corriger,Voir l
es axes,Menu ?"
2510 PEN 2:TEX$="CVM":GOSUB 50000:PEN 1:
FAXY=1
2520 IF K=1 THEN 2000
2530 IF K=2 THEN FTRA=0:GOSUB 3000
2600 RETURN
3000 'TRACE AXE X
3070 MODE 2:INK 0,13:INK 1,0:CLS
3080 LOCATE 15,25:PRINT OX;" ";CHR$(242)
;CHR$(243);" ";MX;" :  ";LGX$
3090 LOCATE 78-LEN(UX$),25:PRINT UX$
3100 LOCATE 13+(65-LEN(TIT$))/2,2:PRINT
TIT$
3110 PLOT 100,48,1:DRAWR 520,0
3120 PLOT 100,48:DRAWR 0,333
3130 FOR I=OXE TO MXE:H=5
3135 VG=VAL(LEFT$(STR$(I),3)):IF I=MXE A
ND MXE>99 AND VG=10 THEN VG=100
3140 C$=RIGHT$(STR$(VG),1):IF C$="5" THE
N H=10
3150 IF C$="0" THEN H=15
3160 PLOT 100+(I-OXE)*RGXE,48:DRAWR 0,H
3170 IF H=15 OR DXE<20 THEN PLOTR -12-4*
(LEN(STR$(VG))-2),-(H+5):TAG:PRINT VG;:T
AGOFF
3180 NEXT
```

```
4000 'TRACE AXE Y
4070 LOCATE 2,2:PRINT UY$:PRINT:PRINT MY
:PRINT:PRINT " ";CHR$(254):PRINT:PRINT O
Y:PRINT
4080 FOR I=1 TO LEN (LGY$):PRINT SPC(5);
MID$(LGY$,I,1)
4090 NEXT
4100 IF DYE>50 THEN ST=2 ELSE ST=1
4110 IF ST=2 AND OYE/2<>INT(OYE/2) THEN
OYE=OYE-1:DYE=DYE+1
4120 FOR I=OYE TO MYE STEP ST:H=5
4125 VG=VAL(LEFT$(STR$(I),3)):IF I=MYE A
ND MYE>99 AND VG=10 THEN VG=100
4130 C$=RIGHT$(STR$(VG),1):IF C$="5" THE
N H=10
4140 IF C$="0" THEN H=15
4150 PLOT 100,48+(I-OYE)*RGYE:DRAWR H,0
4160 IF H=15 OR DYE<20 THEN PLOTR -H-12-
8*LEN(STR$(VG)),4:TAG:PRINT VG;:TAGOFF
4170 NEXT
4180 IF FTRA THEN 4200
4190 CALL &BB06
4200 RETURN
5000 'ENTREE DES DONNEES
5005 WINDOW #0,1,40,7,25:CLS
5010 WINDOW #1,1,40,1,6:CLS #1:BORDER 2
5020 LOCATE #1,2,2:PRINT #1,"ENTREE DES
DONNEES ( Q POUR QUITTER )"
5030 LOCATE #1,2,4:PRINT #1."Num":SPC(13
);"X";SPC(14);"Y"
5040 PRINT #1,STRING$(40." ")
5100 I=1:H=1
5110 LOCATE 1,H:PRINT USING "####":I::PR
INT SPC(13);:INPUT"",D$
5115 IF UPPER$(D$)="Q" THEN NF=I-1:GOTO
5200
5120 IF VAL(D$)=0 AND D$<>"0" THEN PRINT
 CHR$(7):GOTO 5110
5130 PT(I,0)=VAL(D$)
5140 LOCATE 33,H:INPUT "",D$
5150 IF VAL(D$)=0 AND D$<>"0" THEN PRINT
 CHR$(7):GOTO 5140
5160 PT(I,1)=VAL(D$)
```

```
5170 I=I+1:H=H+1
5180 IF I/18=INT(I/18) THEN CLS:H=1
5190 GOTO 5110
5200 WINDOW #0,1,40,1,25:CLS
5210 BORDER 1:PRINT:PRINT "   PATIENCE .
.."
5300 'TRI PAR X CROISSANT
5310 F=0: FOR J= 1 TO NF
5320 IF PT(J,0)>=PT(J-1,0) THEN 5360
5330 FOR R=0 TO 1:PT(0,R)=PT(J,R):PT(J,R
)=PT(J-1,R)
5340 PT(J-1,R)=PT(0,R):NEXT:F=1
5360 NEXT
5370 IF F=1 THEN 5310
5400 RETURN
6000 ' FORMULE
6010 CLS
6020 IF FAXY=0 THEN PRINT CHR$(7);"IL FA
UT D'ABORD DEFINIR LES AXES .....":FOR W
=1 TO 3000:NEXT:RETURN
6030 PEN 3:LOCATE 8,3:PRINT "Le Programm
e est ARRETE :":PEN 1
6040 LOCATE 2,7:PRINT "Entrez la Formule
 en ligne 7000. Ex :"
6050 LOCATE 13,9:PRINT "7000 DEG:Y=SIN(X
)"
6060 LOCATE 2,13:PRINT "ENTER , puis le
. du pave numerique":PRINT
6080 END
6500 PRINT: PRINT "          PATIENCE ...
."
6510 K=1:STX=DX/100:IF INT(STX)=STX THEN
  K=0
6520 I=0:FOR X=OX TO MX+K*STX STEP STX
6530 I=I+1:Z=X
6540 GOSUB 7000
6545 IF OY<0 AND Y<OY*10 THEN Y=OY*10:PR
INT CHR$(7);
6547 IF Y>MY*10 THEN Y=MY*10:PRINT CHR$(
7);
6550 PT(I,0)=Z:PT(I,1)=Y
6560 IF I=1 THEN MINY=Y:MAXY=Y:GOTO 6590
6570 MINY=MIN(Y,MINY)
```

```
6580 MAXY=MAX(Y,MAXY)
6590 NEXT:NF=I
6600 CLS
6610 LOCATE 1,5:PRINT "Les calculs de Y
donnent ces limites :"
6620 PRINT:PRINT "  mini =";MINY
6625 PRINT:PRINT "  MAXI =";MAXY
6630 PRINT:PRINT "Vous aviez prevu de";O
Y;"a";MY
6640 LOCATE 4,20:PRINT "VOULEZ-VOUS REDE
FINIR LES AXES ?"
6650 TEX$="ON":GOSUB 50000
6660 IF K=1 THEN CLS:FTRA=0:GOSUB 2100
6700 GOTO 1000
6990 'FORMULE ENTREE A LA MAIN:
7000 Y=10/X:' ceci est un exemple
7100 RETURN
8000 'TRACE
8010 FTRA=1:GOSUB 3000
8100 PLOT 100+(PT(1,0)-OX)*RGX,48+(PT(1,
1)-OY)*RGY,1
8110 FOR I=2 TO NF
8120 DRAW 100+(PT(I,0)-OX)*RGX,48+(PT(I,
1)-OY)*RGY,1
8130 NEXT:IF FIMP THEN 8200
8150 CALL &BB06
8200 RETURN
9000 'SAUVEGARDE
9010 CLS:PEN 3:LOCATE 16,3:PRINT "SAUVEG
ARDE":PEN 1
9020 LOCATE 11,17:PRINT"( Q = Retour MEN
U )"
9030 LOCATE 13,13:INPUT"NOM: ",FICH$
9035 IF FICH$="" THEN FICH$=TIT$:IF TIT$
="" THEN 9000
9040 FICH$=UPPER$(FICH$):IF FICH$="Q" TH
EN 9500
9050 FICH$=LEFT$(FICH$,8)+".GRA"
9060 OPENOUT FICH$
9070 WRITE#9,TIT$,NF,OX,MX,LGX$,UX$,OY,M
Y,LGY$,UY$
9080 WRITE#9,FAXY,DX,RGX,OXE,MXE,DXE,RGX
E,DY,RGY,OYE,MYE,DYE,RGYE
```

```
9090 FOR I=1 TO NF:WRITE#9,PT(I,0),PT(I,
1):NEXT:CLOSEOUT
9500 RETURN
10000 'CHARGEMENT
10010 CLS:PEN 3:LOCATE 16,3:PRINT "CHARG
EMENT":PEN 1
10020 LOCATE 11,17:PRINT"( Q = Retour ME
NU )"
10030 LOCATE 13,13:INPUT"NOM: ",FICH$
10035 IF FICH$="" THEN 10000
10040 FICH$=UPPER$(FICH$):IF FICH$="Q" T
HEN 10500
10050 FICH$=LEFT$(FICH$,8)+".GRA"
10060 OPENIN FICH$
10070 INPUT#9,TIT$,NF,OX,MX,LGX$,UX$,OY,
MY,LGY$,UY$
10080 INPUT#9,FAXY,DX,RGX,OXE,MXE,DXE,RG
XE,DY,RGY,OYE,MYE,DYE,RGYE
10090 FOR I=1 TO NF:INPUT#9,PT(I,0),PT(I
,1):NEXT:CLOSEIN
10500 RETURN
11000 'IMPRESSION
11010 CLS:FQ=0
11020 LOCATE 5,5:PRINT "SI L'IMPRIMANTE
EST PRETE ..."
11030 LOCATE 10,9:PRINT "D - LISTE DES D
ONNEES"
11040 LOCATE 10,11:PRINT "G - GRAPHIQUE"
11050 LOCATE 10,13:PRINT "Q - QUITTER"
11060 TEX$="DGQ":GOSUB 50000
11070 ON K GOSUB 11200,11400,11600
11075 IF FQ THEN FQ=0:GOTO 11500
11080 GOTO 11000
11200 'LISTE
11205 PRINT#8,CHR$(27);CHR$(64);
11210 PRINT #8,SPC((40-LEN(TIT$))/2);TIT
$:PRINT #8
11220 PRINT #8," Numero";SPC(10);"X";SPC
(14);"Y"
11225 PRINT#8,TAB(16);LGX$;:PRINT#8,TAB(
31);LGY$
11230 PRINT#8,TAB(16);UX$;:PRINT#8,TAB(3
1);UY$:PRINT#8
```

```
11240 FOR I=1 TO NF
11250 PRINT #8, USING "####";I;:PRINT #8
,SPC(8);:PRINT #8,USING "######.###";PT(
I,0);
11260 PRINT #8,SPC(5);:PRINT #8,USING "#
#####.###";PT(I,1)
11270 NEXT:FOR I=1 TO 3:PRINT #8:NEXT
11300 RETURN
11400 'HARD COPY
11410 FIMP=1:GOSUB 8000
11420 CALL &9F00:FIMP=0:PRINT #8,CHR$(27
);CHR$(64);
11430 MODE 1:INK 0,1:INK 1,24
11500 RETURN
11600 'RETOUR MENU
11610 FQ=1
11620 RETURN
12000 ' QUITTER
12010 CLS
12020 END
15000 'INSTRUCTIONS
15010 CLS
15020 PRINT " Ce Logiciel dessine automa
tiquement lesaxes XY avec leurs graduati
ons , suivantles limites que vous aurez
fixees."
15030 PRINT " Les Donnees peuvent etre e
ntrees en de-sordre,il les classera en X
 croissant,oubien par une formule mathem
atique.        Le trace de la courbe est
ultra rapide."
15040 PRINT " On peut imprimer le tablea
u des donneesou le HARD COPY du graphiqu
e.                   Suivre l'ordre logique
suivant:":PRINT
15050 PRINT "1-Definir les caracteristiq
ues des Axes.2-Entrer les donnees ou la
Formule.      3-Trace du graphe.":PRINT
15060 PRINT "  On peut sauvegarder un gr
aphe:Le NOM  de ce fichier de donnees pe
ut etre le   TITRE du graphe si vous rep
ondez par la touche ENTER a la question
NOM."
```

```
15070 PRINT " Si vous venez de recharger
 un fichier- graphe,passez directement a
 l'option T.":PRINT:PRINT "            PRE
SSEZ UNE TOUCHE."
15080 CALL &BB06
15090 RETURN
20000 'TRAITEMENT ERREUR
20010 IF ERL=7000 AND (ERR=6 OR ERR=11)
THEN Y=PT(I-1,1):RESUME 7100
20020 CLS:PRINT "ERREUR NUMERO";ERR;"EN
LIGNE";ERL:END
50000 'REPONSE A UN MENU
50010 LT=LEN(TEX$):R$=""
50020 LOCATE 15-LT,24:PRINT "Reponse (";
50030 FOR I=1 TO LT-1
50040 PRINT MID$(TEX$,I,1);",";:NEXT
50050 PRINT RIGHT$(TEX$,1);")";CHR$(154)
;CHR$(243);CHR$(207)
50060 TEX$=UPPER$(TEX$)
50070 WHILE R$="":R$=INKEY$:WEND
50080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
50090 IF K=0 THEN R$="":PRINT CHR$(7);:G
OTO 50070
50100 RETURN
65535 ' ------ FIN DE LISTING ------
```

# Chapitre IX

# DEFINITION D'UN FICHIER ET DE SES BUTS

Pour le nouveau-venu à la micro-informatique, la première application est l'établissement d'un fichier : il y a le côté spectaculaire de la chose, doublé par la facilité due à de nombreux logiciels du commerce, bon marché, performants et simples d'emploi.
Et c'est là que les bévues s'accumulent ; très souvent on se lance dans ce que l'on appelle vulgairement un ''travail de singe'', c'est-à-dire un travail important mais hélas inutile parce que mal pensé dès le départ.
Première grande règle : *Il faut commencer par la fin !* Disons plutôt par la finalité. Avant de se lancer dans la saisie de centaines de fiches au clavier, il est sage de se poser les questions suivantes :

— La recherche sera-t-elle plus rapide que dans un fichier ''papier'' (fiches bristol, répertoires alphabétiques, etc.), compte tenu qu'il faudra chaque fois mettre l'AMSTRAD sous tension, puis charger le logiciel, puis charger le fichier et enfin faire la recherche ; alors que trouver le numéro de téléphone de DUVAL ne prend que cinq secondes sur votre agenda de poche, ou quarante secondes dans l'annuaire des PTT de votre département... (vrai ou faux ?)
— Deuxième cas, la recherche sur micro (tout compris) prend 5 minutes ; avant, sur papier, il en fallait 30. On semble gagnant ! Pas toujours... La saisie de ce fichier au clavier a demandé 20 heures (pénibles...) ; sa mise à jour obligatoire (eh oui, un fichier doit être ''entretenu'') vous demande au total quatre heures par an. Et maintenant la douche froide : en fait, vous n'y faites en moyenne que cinq recherches par an ! Faites vos comptes ; c'est ridicule. Si, en revanche, vous effectuez dix recherches par mois ou plus, alors là, vive les fichiers en informatique !
— Troisième cas, vous êtes à présent persuadé que l'informatique va supplanter le système papier, mais hélas votre problème personnel, sans être vraiment compliqué, est vraiment très ''spécial''. Vous achetez alors des logiciels de plus en plus sophistiqués et chers, qui

vous proposent des dizaines d'options dont vous n'avez que faire, et qui ne résolvent toujours pas votre problème si particulier. C'est le moment pour prendre conscience que vous appartenez à une minorité ''noble'' : vous êtes un possesseur de micro-ordinateur *sachant programmer.* Faites votre logiciel sur mesures ! C'est très facile pour un gestionnaire de fichiers. Il ne saura traiter que votre problème, certes, mais sur ce point il va ridiculiser par ses performances les plus ''grands'' logiciels du commerce (et c'est gratuit... ).

Encore un argument pour vous lancer dans le ''fait main'' : l'auteur d'un logiciel du commerce ne sait pas du tout ce que veut en faire chacun de ses acheteurs, alors il a multiplié les options, les menus, les sous-menus... Ce désir de ''souplesse'' se traduit par la lourdeur d'emploi, il vous faut toujours donner les mêmes réponses à des cascades de questions, au risque de se tromper... énervant ! Et, bien sûr, il manque le petit détail ''tout bête'' dont vous avez tant besoin. Pire, ce logiciel Diplodocus occupe 25 kilo-octets et il ne vous reste plus grand chose pour y loger votre fichier...
En le programmant vous-même, 75 % de ces questions disparaissent, il ne représente plus que 9000 octets et il fait exactement ce dont vous rêviez. Vous êtes gagnant sur tous les tableaux.

## LE CHOIX DES VARIABLES

Il est aussi dangereux de prévoir trop grand que trop petit. Car passée l'euphorie des vingt premières fiches, il est vexant de se trouver piégé lorsque le fichier dépasse les 10 kilo-octets.
Le premier problème est de déterminer les variables : noms, nombre, longueurs et natures (numériques ou chaînes). Cela s'étudie sur papier, très soigneusement, même si plusieurs heures sont nécessaires. C'est une succession de cas de conscience car il faut être à la fois radin et prévoyant ! Nous serons plus directs en disant qu'il faut uniquement conserver les variables (rubriques) pouvant être des *''voies d'accès'',* plus quelques variables vraiment indispensables. Une fiche ne doit pas être une ''fiche signalétique'' complète, mais une pièce de repérage.
Un exemple concret : supposons que vous êtes le vendeur de voitures d'occasion d'un grand garage : les acheteurs vont vous demander ''qu'avez-vous comme RENAULT ?'', ''comme BREAK ?'', ''comme moins de 6 CV ?'', ''comme voitures rouges ?''. Que de voies d'accès au fichier ! Marque, carrosserie, puissance, couleur, modèle, année, kilométrage, prix. Autant de variables de sélection, plus une autre indispensable où se trouve tout ce qui concerne cette voiture (clés, carte grise, emplacement dans le garage, etc.). Dans ce fichier informatique, il serait inutile et encombrant de faire figurer des variables telles que numéro minéralogique, ancien propriétaire, date d'entrée, etc.).
Il reste maintenant à déterminer les longueurs maxi pour chaque rubrique ; c'est l'occasion de prévoir une codification stricte des valeurs entrées. Exemple : rubrique couleur, on aura six caractères pour blanc,

noir, jaune, bleu, rouge, vert, violet, marine, gris, ciel beige, brun, orange. Interdiction d'entrer des couleurs telles que ocre, paille, car la fiche ne sortira pas si on recherche jaune : peu importe si la fiche n'est pas très précise, l'essentiel est que le fichier soit efficace. Ne perdez jamais cela de vue ! Un fichier informatique n'est pas une pièce d'archive bibliographique, c'est une chose vivante, faite pour être triturée dans tous les sens.

Un autre problème concernant les longueurs maxi est l'éventualité d'édition sur imprimante sous forme de tableau : si votre imprimante ne peut faire que 80 caractères par ligne, il faut tenir ses comptes ! = Total des longueurs + 1 caractère par rubrique (séparation mini). Pensez aussi au fait que la longueur de la légende ne peut être supérieure à la longueur maxi prévue. Exemple : un caractère vous suffit pour la variable ''TYPE'', mais sur l'en-tête du tableau, vous ne pourrez placer que le ''T'' ; pas très clair... Alors, portez la longueur maxi à 2, car ''TY'' sera compréhensible.

Vous pouvez alors aborder les considérations globales, à savoir l'estimation du nombre maximum de fiches possibles : comptez 20000 octets divisé par le nombre d'octets réservés pour une fiche. Exemple : un tableau DIM de chaînes totalise 57 caractères par ''ligne'' ; 20000/57 ≅ 350 fiches. Pour des DIM numériques, comptez deux octets pour des ''entiers'' et 5 octets (deux fois et demi plus !) pour des ''réels''. Dites-vous bien que des rubriques, remplies ou non, occupent pratiquement la même place en mémoire.

Pour notre vendeur de voitures d'occasion, ce maximum de 350 fiches ne sera pas un problème (on le lui souhaite...) ; en revanche, s'il s'agit de gérer une bibliothèque qui approche déjà les 2000 ouvrages, il faudra envisager un fichier par genre (policiers, voyages, etc.) ou groupant quelques ''petits'' genres.

Le nombre, la nature, la longueur maxi des variables constituent ce que l'on appelle la *structure* d'un fichier. Dans un logiciel du commerce, il est la plupart du temps impossible de la modifier par la suite. Heureusement, c'est *toujours* possible avec un logiciel ''fait maison'', en transférant le contenu d'un tableau DIM dans un autre de dimensions différentes (voir MPSA chapitre XII).

## LE CHOIX DU MENU

De nombreuses options classiques dans un logiciel du commerce vont disparaître, parce que vous les avez fixées dans le programme. C'est le cas de :

— Création du fichier : Toute la structure (DIM) ainsi que les noms des variables sont déjà dans vos lignes Basic = un gain de 3000 octets de Basic.
— Chargement du fichier : Il peut être automatique dès le lancement par RUN. Même le nom du fichier à charger est dans le programme.

Il vous faut maintenant prévoir les différentes ''manœuvres'' qui vous seront réellement utiles dans la pratique. Voilà qui va supprimer bien des questions en sous-menus. Exemple : le tri (classement). En avez-vous vraiment besoin ? Si oui, sur quelle variable ? Ceci étant déterminé dans le programme, il vous suffira de demander l'option TRI au menu principal pour que cela se fasse tout seul sans autres questions. Près de vingt lignes Basic de gagnées. Idem pour les autres options (Recherche, Edition sur imprimante, etc.).
Le résultat final sera le suivant : un programme très complet faisant moins de 10 kilo-octets (autant de gagné pour le fichier). Un emploi simple donc rapide et sans ambiguités (beaucoup moins de questions à l'écran). Adjonction d'options généralement absentes dans les logiciels du commerce, par exemple enregistrement d'un fichier rassemblant une sélection de fiches, ou encore impression bien calibrée sur des étiquettes auto-adhésives (étiquettes mailing), etc.
Et pour un autre genre de fichier ? Vous chargez votre logiciel, vous supprimez des blocs de lignes, vous en modifiez, ajoutez d'autres et vous faites SAVE sous un autre nom de logiciel. C'est tellement rapide à faire que l'auteur en a déjà une vingtaine sur des sujets les plus divers. Un exemple : le plus simple d'entre eux, celui qui permet d'imprimer la liste des variables d'un programme Basic, en ordre alphabétique, telle celle relative à MACHDIM (chapitre IV). Cette liste est aussi sauvegardée sur disquette sur le nom MACHDIM.LIS. Cette extension ''.LIS'' est donnée par le programme (''LISTVAR.BAS''). Moralité : simplicité + spécificité + rapidité + clarté = efficacité. Un aveu : ce programme a été réalisé en moins d'une demi-heure à partir d'un autre !
Ce chapitre préliminaire sur les gestionnaires de fichiers avait deux grands buts, vous faire prendre conscience que :

1 — Définir les structures optimales d'un fichier est une étude longue et détaillée. Le temps passé à cela sera rapidement amorti, et récompensé par une efficacité spectaculaire et durable.
2 — Il faut laisser les logiciels du commerce à ceux qui ne savent (ou ne veulent) pas programmer. En faisant un logiciel adapté au genre du fichier et à son usage futur, on est gagnant sur tous les tableaux.

**NOTE** : Nous verrons au chapitre XV comment récupérer un fichier obtenu avec un logiciel du commerce (sans le déprotéger), afin de poursuivre son exploitation par un logiciel personnel.

## COMMENT ALLONS-NOUS PROCEDER ?

Un gestionnaire de fichiers est obligatoirement en ''programmation structurée''. Chaque module correspond à une option (saisie, recherche, etc.). Tout part du MENU et revient ensuite à ce menu d'options, pour repartir vers d'autres options. De ce fait, on appelle parfois ''Pro-

gramme Talon'' la partie du listing (vers le début) contenant le menu principal.

Le chapitre qui suit concerne un cas particulier, une technique très spéciale qui ne s'adresse qu'aux possesseurs de lecteurs de disquettes, et de préférence aux AMSTRAD CPC 664 et 6128 plutôt qu'à l'AMSTRAD CPC 464.
Les chapitres suivants traiteront chacun d'un module. A titre d'exemple, il s'agira d'un logiciel pour gérer la liste de vos... logiciels ! Il vous sera ensuite facile d'en enregistrer une version modifiée pour tout autre fichier.

# Chapitre X

# L'APPEL EN CHAIN MERGE

Si vous ne possédez pas de lecteur de disquettes, vous pouvez passer au chapitre suivant. Si vous avez un CPC 464 équipé d'un ''drive'', cette technique est possible, chaque appel d'option va demander 25 secondes (un bug du 464) au lieu de 3 à 4 secondes sur les CPC 664 et CPC 6128.
L'intérêt de cette pratique est de disposer de plus de 30 kilo-octets pour le fichier, même si le programme représente plus de 40 kilo-octets !
L'astuce est la suivante : le ''programme talon'' reste constamment en mémoire RAM ; lorsqu'un module-option est appelé, il est lu sur la disquette et vient se greffer à la suite du talon. En fin d'usage, il est effacé et c'est un autre module-option qui vient se greffer au ''talon''.
Le fichier reste en permanence en mémoire. Conséquences :

1 — La partie programme en mémoire dépasse rarement 3000 octets seulement.
2 — Le fichier en mémoire peut être énorme (>30000 octets).
3 — L'ensemble des modules-programme sur la *même face de disquette* peut très bien dépasser les 100 kilo-octets...
4 — Seul inconvénient : le chargement-greffe (= CHAIN MERGE) d'un sous-programme n'est pas aussi instantané que l'habituel accès par un GOSUB. Sauf pour les virtuoses du 6128 qui logeront tous ces petits programmes en ''disque virtuel'' dans les 64 K supplémentaires ; mais ceci sortirait du cadre de cet ouvrage.

## LA COMMANDE BASIC ''CHAIN MERGE''

Vous connaissez déjà la commande MERGE ''nom de programme'' qui ajoute au programme en mémoire le listing du programme ainsi appelé. CHAIN MERGE fait plus : les variables du programme en cours *restent en mémoire* et le programme se poursuit automatiquement sur ce qui vient d'être greffé. Une sorte de GOTO la suite. Programmes Talon + greffon forment un bloc.

**SUPER IMPORTANT** : Tous les modules appelables doivent commencer par le MEME NUMERO DE LIGNE ; par exemple 5000. De même, il est très prudent d'effacer l'ancien greffon (DELETE 5000 – ) avant d'en greffer un autre.
Si les variables sont conservées, ce n'est pas le cas des ''pointeurs'' de numéros de ligne (GOSUB, GOTO, ON ERROR GOTO, etc.), idem pour les DATA. D'où ces précautions :

- Ne pas utiliser de ON ERROR GOTO, ni ON BREAK GOSUB.
- Si vous avez fait un READ sur *tous* les DATA du programme talon et que vous voulez lire des DATA au sein d'un greffon, il est prudent de confirmer par un RESTORE (exemple RESTORE 5050) ; sinon le READ risque de repartir du beau milieu des DATA du talon, pourtant déjà lues ! (Pour l'emploi rationnel des DATA, voir MPSA chapitre XII.)

## LE PROGRAMME TALON

Le mot ''talon'' a plusieurs synonymes selon les auteurs ou traducteurs ; vous trouverez aussi ''souche'', ''primaire'', ''permanent'', ''Base'' ou même ''Mother File''...
Ce programme comprend trois parties successives :

**• L'initialisation**
On n'y passe qu'une seule fois. C'est la définition des DIM, DEFINT, MEMORY, etc. La page d'écran-titre, bien sûr facultative, des définitions de SYMBOL, KEY. Et pourquoi pas le chargement du fichier à traiter ou à compléter.

**• Le menu principal**
Prenez la bonne habitude de le faire commencer à la ligne 1000, et ce dans *tous* vos programmes, car c'est super pratique. La fin d'exécution d'une option se termine toujours par un GOTO 1000. En conséquence, le module menu doit toujours débuter par une ré-initialisation des paramètres d'affichage, par exemple :

```
1010 MODE 1:BORDER 1:PEN 1:PAPER 0:CLS
```

• Le CHAIN MERGE du sous-programme choisi.
• Une ligne bidon (un REM) dont le numéro est celui par lequel commencent tous les sous-programmes ; dans notre exemple 5000, donc programmez en final 5000'.
A titre d'exemple, nous avons rédigé un programme très simple, voire simpliste, car il s'agit d'une démonstration plutôt que d'un utilitaire. Trois rubriques : nom, prénom, date de naissance mais en format AAMMJJ (860907 = 7 septembre 86).

```
10 'FICHMERG - LOGICIEL EN CHAIN MERGE
20 'PROGRAMME TALON (PERMANENT EN RAM)
50 OPENOUT "BIDON":MEMORY HIMEM-1:CLOSEO
UT
60 DIM PROG$(15):'NOMS DES SOUS-PROGR.
70 DIM FIC$(2000,2):'FICHIER A GERER
80 'INDICES:0=NOM ;1=PRENOM ;2=DATE EN F
ORMAT AAMMJJ
90 DATA CHARG,SAISIE,VISION,RECHER,SAUVE
100 FOR I=1 TO 5:READ PROG$(I):NEXT
110 NF=0:'NOMBRE DE FICHES EN RAM
1000 'MENU ( ET POINT DE RETOUR )
1010 BORDER 9:MODE 1:INK 0,1:INK 1,24:IN
K 2,20:INK 3,6:PAPER 2:CLS
1020 LOCATE 16,2:PEN 3:PRINT "M E N U";S
PC(4);NF;"fiches"
1030 PEN 0:LOCATE 10,5:PRINT "C - CHARGE
MENT"
1040 LOCATE 10,7:PRINT "S - SAISIE"
1050 LOCATE 10,9:PRINT "V - VISIONNER"
1060 LOCATE 10,11:PRINT "R - RECHERCHE"
1070 LOCATE 10,13:PRINT "E - ENREGISTREM
ENT"
1080 LOCATE 10,15:PRINT "Q - QUITTER"
1100 TEX$="CSVREQ":GOSUB 3000
1110 IF K=6 THEN BORDER 1:PAPER 0:PEN 1:
CLS:DELETE 5010-:END
1120 CLS:LOCATE 12,12:PRINT "PATIENCE ..
."
1130 CHAIN MERGE PROG$(K),5000,DELETE 50
00-
1140 END
3000 'REPONSE A UN MENU
3010 LT=LEN(TEX$):R$=""
3020 LOCATE 15-LT,24:PRINT "Reponse (";
3030 FOR I=1 TO LT-1
3040 PRINT MID$(TEX$,I,1);",";:NEXT
3050 PRINT RIGHT$(TEX$,1);")";CHR$(154);
CHR$(243);CHR$(207)
3060 TEX$=UPPER$(TEX$)
3070 WHILE R$="":R$=INKEY$:WEND
3080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
```

```
3090 IF K=0 THEN R$="":PRINT CHR$(7);:GO
TO 3070
3100 RETURN
5000 'LIGNE BIDON OBLIGATOIRE
```

En lignes 3000, vous reconnaissez notre routine ''Réponse à un menu'' qui d'habitude est en 50000 ; nous avons fait MERGE ''MENU'' puis RENUM 3000, 50000, car tous les numéros de lignes doivent être inférieurs à 5000.
La ligne 1130 contient le CHAIN MERGE suivi de ses paramètres. Paramètres obligatoires et non facultatifs comme le prétend à tort le manuel du CPC 464.
Le ''vrai'' END est celui de la ligne 1110 car le programme ne peut pas passer par la ligne 1140, laquelle est là pour la clarté du listing et comme sécurité secondaire.

— Avant d'attaquer les sous-programmes, mentionnons deux points importants :
  a) ils se terminent tous par GOTO 1000,
  b) les possesseurs d'AMSTRAD CPC 464 devront les enregistrer en ASCII par SAVE''nom'',A. Sinon plantage probable en chargement avec le message ''EOF MET'' : un bug du 464 qui fait que le chargement d'un octet égal à 26, lors d'un MERGE ou CHAIN MERGE d'un programme Basic, est traduit par CTRL Z qui est le repère de fin de fichier d'un fichier ASCII. Ce bug a été corrigé sur les 664 et 6128 et peut-être le sera-t-il sur les versions plus récentes du 464 (?).

## LE MODULE SAISIE

Très simple et très court, promis ! Les trois rubriques de chaque fiche sont saisies sur la même ligne d'écran. Le CHR$(11) des lignes 5050 et 5060 fait remonter le curseur d'un cran pour rester sur la même ligne. Pas de contrôle d'entrée ni de rattrapage d'erreur, mais les valeurs manquantes sont remplacées par un point (lignes 5070, 5080).

```
5000 'SS PROGRAMME SAISIE
5010 MODE 2:INK 0,20:INK 1,0:PAPER 0:PEN
 1:BORDER 16:CLS
5020 LOCATE 20,2:PRINT "ENTRER NOM PRENO
M DATE(AAMMJJ) Q=QUITTER":PRINT
5030 PRINT USING"####";NF+1;:INPUT" ",FI
C$(NF+1,0)
5040 IF FIC$(NF+1,0)="Q" THEN 5200
```

```
5050 NF=NF+1:PRINT TAB(40);CHR$(11);:INP
UT"",FIC$(NF,1)
5060 PRINT TAB(70);CHR$(11);:INPUT"",FIC
$(NF,2)
5070 IF FIC$(NF,1)="" THEN FIC$(NF,1)=".
"
5080 IF FIC$(NF,2)="" THEN FIC$(NF,2)=".
"
5100 GOTO 5030
5200 GOTO 1000
```

## LE MODULE VISION

Pour visionner à l'écran ou à l'imprimante les fiches à partir de tel numéro. Programme très élémentaire, mais remarquez le GOSUB 3000 de la ligne 5040 : on a le droit puisque l'on est entre deux CHAIN MERGE, au sein d'un listing compact.

```
5000 'SS PROG. VISION
5010 MODE 2:INK 0,13:INK 1,0:PAPER 0:PEN
 1:BORDER 17:CLS
5020 PRINT:INPUT "  A PARTIR DE QUEL NUM
ERO ";NUM$:NUM=VAL(NUM$):IF NUM>NF THEN
5000
5030 PRINT:PRINT "  SUR ECRAN OU IMPRIMA
NTE ?"
5040 TEX$="EI":GOSUB 3000:D=0:IF K=2 THE
N D=8
5050 CLS:PRINT #D,TAB(10);"FICHIER ";NOM
$:PRINT #D
5060 FOR I=NUM TO NF
5070 PRINT #D,USING"####";I;:PRINT #D,"
";FIC$(I,0);:PRINT #D,TAB(40);FIC$(I,1);
:PRINT #D,TAB(70);FIC$(I,2)
5080 NEXT
5090 PRINT:PRINT "  PRESSEZ UNE TOUCHE":
CALL &BB06
5100 GOTO 1000
```

## LE MODULE RECHERCHE

Toujours dans le même style. On demande la rubrique, puis la valeur à rechercher. La ou les fiches sont affichées à l'écran.
A noter que la recherche s'effectue avec la fonction INSTR (ligne 5100), beaucoup plus prudent que le signe =.
Peu de choses à ajouter à ce sous-programme pour qu'il devienne très ''sérieux'' :
une sécurité pour bien demander un numéro de fiche compris entre 1 et NF,
— si rubrique = date, proposer >, < ou = et opérer sur le VAL de cette rubrique.

```
5000 'SS PROG.RECHER
5010 MODE 2:INK 0,20:INK 1,1:PAPER 0:PEN
 1:BORDER 0:CLS
5020 LOCATE 20,2:PRINT "RECHERCHE DANS Q
UELLE RUBRIQUE ?"
5030 PRINT:PRINT " Indice , Nom , Prenom
 , Date ou Quitter"
5040 TEX$="INPDQ":GOSUB 3000
5050 IF K=5 THEN 5300
5060 C=K-2:LOCATE 10,10:INPUT"VALEUR REC
HERCHEE : ",X$:CLS:PRINT
5070 IF K=1 THEN N=VAL(X$):GOSUB 5200:GO
TO 5120
5080 FOR N=1 TO NF
5090 IF K=1 THEN N=VAL(X$):GOSUB 5200:GO
TO 5110
5100 IF INSTR(FIC$(N,C),X$) THEN GOSUB 5
200
5110 NEXT
5120 PRINT CHR$(7):PRINT "    FIN DU FICH
IER  -  PRESSEZ UNE TOUCHE"
5130 CALL &BB06:GOTO 5000
5200 'AFFICHAGE
5210 PRINT USING "####";N;:PRINT " ";FIC
$(N,0);:PRINT TAB(40);FIC$(N,1);:PRINT T
AB(70);FIC$(N,2)
5220 RETURN
5300 GOTO 1000
```

## LES MODULES SAUVEGARDE ET CHARGEMENT

Ils sont très courts et ressemblants. Pour un logiciel plus étoffé, il ne manquerait qu'une sécurité sur la syntaxe du nom de fichier.
Autant dire qu'ils resteront si courts qu'ils pourraient être intégrés dans le programme talon.

```
5000 'SS PROG. SAUVE
5010 PAPER 0:PEN 1:BORDER 25:CLS
5020 LOCATE 8,10:PRINT "ENREGISTREMENT D
U FICHIER"
5030 LOCATE 10,14:INPUT "NOM : ".NOM$
5040 OPENOUT NOM$:WRITE#9.NF
5050 FOR I=1 TO NF
5060 FOR J=0 TO 2
5070 WRITE #9,FIC$(I,J):NEXT:NEXT
5080 CLOSEOUT
5100 GOTO 1000
```

```
5000 'SS PROGRAMME CHARG
5010 BORDER 9:PAPER 0:PEN 3:CLS
5020 LOCATE 9,6:PRINT "CHARGEMENT D'UN F
ICHIER:"
5030 PEN 2:LOCATE 10,12:INPUT"NOM : ".NO
M$
5040 OPENIN NOM$:INPUT#9,NF
5050 FOR I=1 TO NF:FOR J=0 TO 2
5060 INPUT#9,FIC$(I,J):NEXT:NEXT
5070 CLOSEIN
5080 GOTO 1000
```

## CONCLUSION

Vous avez remarqué que nous changeons les couleurs d'un module à l'autre ; ce n'est pas pour faire carnaval mais c'est une mesure contre les étourderies de l'opérateur.
Ce programme, bien qu'étant très court, fonctionne parfaitement ; il lui manque bien des options telles que le tri, les modifications de fichiers, les sélections, etc. Nous ne voulions qu'une ''démo'' pour cette technique adoptée pour les gros logiciels professionnels, réservée soit pour les très gros logiciels ou pour les très gros fichiers.

# Chapitre XI

# LANCEMENT D'UN GESTIONNAIRE DE FICHIERS

Cette partie comprend la définition de la structure du fichier et du programme, ainsi que le menu principal.
Notre logiciel baptisé ''LOGICLAS'' est destiné à rassembler les titres et caractéristiques de logiciels pour AMSTRAD. En fait, il est écrit de telle sorte qu'il n'y a que quelques lignes à modifier pour l'adapter à tout autre domaine. Le listing complet représente près de 8800 octets, ce qui est très peu eu égard des performances et de sa souplesse d'utilisation. Il surpasse très largement des logiciels du commerce faisant plus de 15 kilo-octets. Pourquoi ? Une multitude d'astuces de programmation, que nous expliquerons au fur et à mesure, afin que vous puissiez modifier ou transposer des passages ; car, répétons-le, il est aussi conçu pour être un LOGICIEL de BASE.
Pour ''aérer'' ce livre, nous avons éclaté la description de ce programme sur quatre petits chapitres.

## LA STRUCTURE (Lignes 10 à 270)

Les deux premières lignes de REM sont très importantes pour le futur : le nom, le rôle, le micro concerné et la date ; non pas de la création mais de la *dernière retouche.*
Sans ces deux lignes, imaginez-vous des mois, des années plus tard, en découvrant plusieurs versions de ce listing... Donc première consigne : de l'ordre !
La ligne 30, très classique, nous met à l'abri d'un défaut AMSTRAD : la ''descente'' de la zone des variables à chaque OPENOUT et OPENIN. Le MEMORY est tout de suite abaissé à sa valeur mini, ce qui nous évitera bien des ennuis.

La ligne 40 décide que toutes les variables numériques sont des ''entiers'', sauf celles dont le nom commence par Z. Deux avantages : gain en mémoire (2 octets au lieu de 5) et boucles FOR NEXT deux fois plus rapides. On confirme les INK d'origine, une sage précaution. Dans un programme ordonné, toutes les définitions de DIM doivent être au début du programme (une prudence) et sur une même ligne car c'est plus pratique. Ce numéro de ligne sera 50 ou 100 parce qu'il est facile de s'en souvenir.

```
10 ' LOGICLAS - GESTION DE LOGITHEQUE
20 ' AMSTRAD CPC - Michel ARCHAMBAULT /
4.1986
30 OPENOUT "BIDON":MEMORY HIMEM-1:CLOSEO
UT
40 DEFINT A-Y:INK 0,1:INK 1,24:INK 2,20:
INK 3,6
50 NRU=8 : ' NOMBRE DE RUBRIQUES
100 DIM F$(300,NRU),S(300),LR(NRU)
110 DATA TITRE,CODEDISC,GENRE,EDITEUR,AU
TEUR,DISC,K7,NOTICE,REM
120 FOR R=0 TO NRU:READ F$(0,R):NEXT
130 DATA 21,8,12,15,15,7,5,7,15
140 FOR R=0 TO NRU:READ LR(R):NEXT:'Long
ueurs Rubriques
200 ' ADRESSES DES MODULES
210 '    2000 SAISIE
220 '    3000 RECHERCHE SELECTION
230 '    4000 MODIFICATION EFFACEMENT
240 '    5000 TRI
250 '    6000 IMPRESSION
260 '    7000 SAUVEGARDE
270 '    8000 CHARGEMENT
1000 ' MENU
1010 MODE 1:INK 1,24:PAPER 0:PEN 3:BORDE
R 1:CLS
1020 LOCATE 11,2:PRINT "L O G I T H E Q
U E":PEN 2
1030 LOCATE 30,4:PRINT NF;"fiches":PEN 1
1040 DATA SAISIE,RECHERCHE-SELECTION,MOD
IFICATION-EFFACEMENT,TRI,IMPRESSION,ENRE
GISTREMENT,DISC-UTIL,CHARGEMENT,QUITTER
1050 RESTORE 1040:FOR I=1 TO 9:READ OPT$
```

```
1060 LOCATE 10,I*2+4:PRINT LEFT$(OPT$,1)
;" - ";OPT$:NEXT
1070 TEX$="SRMTIEDCQ":PEN 2:GOSUB 50000
1080 ON K GOSUB 2000,3000,4000,5000,6000
,7000,60000,8000,1100
1090 GOTO 1000
1100 BORDER 1:MODE 1:PAPER 0:PEN 1:CLS
1110 END
```

**Les DIM**

Le fichier sera en mémoire dans le tableau F$ (F comme fiche) ; 300 fiches maxi avec 9 rubriques numérotées de 0 à 8. C'est une dimension (300 × 9) qu'il est rare de dépasser dans la pratique dans d'autres domaines. Avec moins de rubriques, on pourrait augmenter le nombre de fiches.

Le tableau S(300) est numérique, un nombre entier par ''ligne''. S veut dire sélection, c'est dans celui-ci que nous placerons les *numéros* des fiches sélectionnées par tel critère. Pourquoi 300 lignes réservées alors que moins serait plus réaliste pour une sélection ? Parce que ce DIM va aussi nous servir au tri du fichier ; un tri ''turbo'' dont nous parlerons plus loin.

Le tableau LR(8) est numérique ; il va contenir les Longueurs maxi des Rubriques de 0 à 8. En effet, pour éviter des gaspillages de mémoire, il faut calibrer les variables chaîne d'un tableau DIM.

**Les Rubriques**

Elles sont dans les DATA de la ligne 110. Et nulle part ailleurs. Il va donc être facile et sûr de changer leurs noms, par exemple pour un autre logiciel, ou pour changer un nom de rubrique qui ne vous semble pas très explicite.

Pour ce logiciel, nous avons défini ceci :

— TITRE (21 caractères), c'est le nom commercial.
— CODEDISC (8 caractères) est le nom d'appel sur disquette.
— GENRE (12 caractères). Arcade, DAO, Aventure, etc.
— EDITEUR (15 caractères).
— AUTEUR (15 caractères).
— DISC (7 caractères) Ex.: 12B = disc n° 12 face B.
— K7 (5 caractères) = cassette (original, numéro, etc.).
— NOTICE (7 caractères) F = Français ; A = Anglais, IN = Incluse.
— REM (15 caractères) Remarques diverses.

Ces longueurs maxi ont fait l'objet d'une étude très méthodique ; ainsi 21 pour le titre car cela correspond au nom répertorié le plus long (''WAY OF EXPLODING FIST''). De plus, il faut vérifier que le total de ces valeurs plus le numéro de fichier, plus deux caractères pour séparer chaque rubrique, ne dépasse pas la largeur maxi pour notre imprimante... On arrive ainsi à 130.

Il faut également veiller à ce que le nom de rubrique ne soit pas plus long que sa longueur maxi, à cause des têtes de colonnes sur les tableaux d'impression.
Ces noms vont être mis en ligne zéro du tableau DIM F$. Mais cette ligne zéro ne sera pas sauvegardée avec le reste du fichier ; ainsi on garde le loisir de modifier ces noms en ligne 110, sans rien toucher au fichier existant. Chose le plus souvent impossible avec les logiciels du commerce.

## LE CHANGEMENT DE STRUCTURE

Vous désirez à présent construire un tout autre fichier, par exemple pour gérer vos enregistrements de films sur cassettes vidéo. Oh ! que c'est rapide !
- Changez le nom en lignes 10 et 1020 et la date en ligne 20.
- Changez le nombre de rubriques ligne 50.
- Eventuellement le nombre de fiches maxi dans les DIM de la ligne 100 (ici = 300).
- Changez les noms des rubriques ligne 110 et leurs longueurs maxi, ligne 130.
- Le TEX$ des menus de rubrique.

**IMPORTANT** : Les numéros de rubriques vont de zéro à NRU. Donc pour 9 rubriques NRU = 8 (ligne 50).

Vous constatez qu'il faut moins de dix minutes pour concevoir un "frère" au logiciel "LOGICLAS", et à la sauvegarde sous un autre nom ; puisque tout est paramétré.
Si vous avez besoin de rubriques numériques, prévoyez en ligne 100 un DIM numérique "parallèle" au DIM F$. Autre solution : entrez ces nombres en DIM F$, en sachant que pour la recherche ou le tri, il faudra considérer le VAL de ces rubriques.

## LES ADRESSES DES MODULES
## (Lignes 200 à 270)

Ces REM constituent une table des matières très précieuse. Ce sont les numéros de ligne où commencent les divers sous-programmes. Si on veut modifier le module d'édition, on a plus vite fait de taper LIST – 1000, que de rechercher le listing pour y découvrir que c'est à partir de 6000...

## LE MENU

Tradition oblige, toujours en ligne 1000. N'oubliez pas que c'est le point des départs et retours. Donc on commence par confirmer les paramè-

tres ''standard'' d'affichage, en ligne 1010. En principe, lorsqu'ils sont modifiés dans un sous-programme (et c'est le cas), on remet les ''choses en place'' juste avant le RETURN final de ce module. Mais lors de la mise au point ou d'un imprévu, le programme s'y arrête ; on repart alors par l'éternel GOTO 1000 et sans cette ligne 1010, ce ne serait pas triste ! Exemple : lettres noires sur fond bleu marine quand on vient de la saisie...
La ligne 1030 fait afficher en haut à droite le nombre total de fiches en mémoire (NF). Ce n'est pas un gadget, c'est indispensable.
Les trois lignes 1040 à 1060 vont afficher la liste des options précédées de leur lettre initiale. Plusieurs choses à remarquer :
— les noms sont en DATA (modifications faciles),
— le RESTORE 1040 de la ligne 1050 ne sert pas la première fois ; obligatoire pour tous les GOTO 1000 suivants,
— le LOCATE de la ligne 1060 est paramétré. Si on ajoute ou supprime une option, cette ligne ne change pas,
— la lettre initiale est ''sortie'' par un LEFT$ (ligne 1060).

On ne présente plus le TEX$ et le GOSUB 50000 de la ligne 1070 (chapitre II).
Après les GOSUB de la ligne 1080, on tombe sur le GOTO 1000 qui redonne le menu en fin de chaque option.
L'option ''quitter'' (lignes 1100-1110) réinitialise les paramètres d'affichage et fait un CLS + END. On a alors la main pour faire n'importe quoi (sauvegarde du programme, listing, etc.), tout en conservant le fichier en RAM. On peut repartir par GOTO 1000.

**Le nom des options**
On doit veiller à ce que chaque nom commence par une lettre différente. Ne présentez jamais une liste d'options en les numérotant de 1 à ... Cela est pénible pour l'opérateur qui perd du temps à ''décoder'' le menu. Après il croit se souvenir du nombre à taper et c'est le risque d'erreur.
Nous avons mis ensemble des actions différentes telles que Recherche et Sélection ou Modification et Effacement. Ne croyez pas que c'était pour gagner deux lignes ! C'est parce que dans l'utilisation pratique, ces actions sont menées conjointement, alternativement. Il serait donc lourd de repasser alors par le menu principal.
Veillez également à ce que les options les plus fréquemment appelées soient *en haut* du menu. C'est pour cela que ''Chargement'' est tout en bas, car on ne le demande qu'une seule fois, au début.
Le choix des couleurs a aussi son importance. Ici, le titre ''LOGITHEQUE'' est en rouge sombre sur fond marine, peu lisible, de même que la liste des codes en bleu ciel sur la ligne 24. En revanche, la liste des options ''crache'' en jaune sur ce fond. Or c'est là que l'opérateur cherche ce qu'il veut ; il est donc ''guidé'' par l'écriture voyante.
Vous constatez que la combinaison, la somme de menus détails d'aspect anodin, va rendre votre programme sympathique à l'utilisateur.

Un menu principal clair et pratique est une chose primordiale puisque l'on y revient souvent.
Vous allez dire ''je ne vais pas commercialiser mon programme : l'utilisateur, c'est moi !''. Je vous répondrai ''raison de plus'' ; on n'est jamais aussi bien gaté que par soi-même.

## CONCLUSION

Faites un bref bilan de tout ce que l'on vient de définir et de mettre en marche, et maintenant considérez combien le listing est court. Les boucles FOR NEXT, les DATA et READ y sont pour beaucoup ; et en plus, cela nous confère quelque chose de facilement remodelable.
Lorsque l'on débute en Basic, on a trop tendance à programmer des suites de PRINT, parce que l'on a l'habitude du stylo... Alors que si on se donne la peine de cogiter un tout petit peu, tout se fait en deux ou trois lignes (voir lignes 1040, 1050, 1060).

```
        L O G I T H E Q U E
                               213 Fiches

      C - CHARGEMENT
      S - SAISIE
      R - RECHERCHE-SELECTION
      M - MODIFICATION-EFFACEMENT
      T - TRI
      I - IMPRESSION
      E - ENREGISTREMENT
      D - DISC-UTIL
      Q - QUITTER
  Reponse (C,S,R,M,T,I,E,D,Q)
```

# Chapitre XII

# SAISIE, MODIFICATION, EFFACEMENT DE FICHES

L'écran de saisie des données est celui sur lequel l'opérateur passe le plus de temps ; en conséquence les erreurs par étourderie vont devenir fréquentes au bout d'un quart d'heure si la ''manipulation'' demande trop d'attention.
On doit pouvoir *se passer de regarder l'écran,* seulement les notes lues qu'il faut entrer au clavier. Autrement dit, l'opérateur, pour se concentrer sur ses notes, doit avoir confiance en la machine :
1) qu'il ait peu de risques de se tromper,
2) que si cela arrive, la retouche soit facile.
Un tel tableau de saisie, est-il de l'utopie ? Le nôtre prouve le contraire.

## L'ECRAN DE SAISIE

Fond bleu ciel, en MODE 1. En haut, le numéro de la fiche en cours ; c'est donc une page d'écran par fiche.
Au-dessous, une ligne par rubrique ; nom de la rubrique suivi par l'espace d'écriture délimité par le signe >.
Ces noms de rubriques ont des longueurs différentes, mais les zones d'écriture partent toutes de la même tabulation horizontale. Couleurs des lettres, noir.
Sous les rubriques, deux autres lignes, en rouge : ''SUIVANTE'' et ''QUITTER''.
Au départ, nous avons donc cette page de saisie avec le curseur positionné au début de la première rubrique.

Début de saisie : on tape ENTER après chaque inscription, mais, et ce point est intéressant, on peut descendre le curseur par la flèche ''bas'' aussi bien que par ENTER. On peut aussi remonter par la flèche ''haut'', pour corriger une rubrique déjà remplie. Faire ENTER sur une rubrique déjà écrite ne la modifie pas.

Le secret de ce confort est le suivant : c'est la *position verticale du curseur* (fonction Basic VPOS) qui détermine à quelle rubrique correspond la chaîne que vous venez d'entrer ; tout simplement.

**IMPORTANT** : Si vous corrigez, vous devez retaper tout le contenu de la rubrique, car c'est un LINE INPUT (ligne 2030), lequel vous autorise les virgules, blancs au départ, guillemets, etc.
Si vous n'avez rien à écrire dans plusieurs rubriques successives, vous gagnerez du temps en descendant avec la flèche ''bas''.
Si vous arrivez sur la ligne rouge ''suivante'' à l'aide de la touche ENTER, un beep sonore vous prévient qu'au prochain coup il passera à la fiche suivante, ce qui vous invite à vérifier notre page d'écran : c'est pour la saisie rapide sans regarder l'écran (ligne 2035 : faites-la sauter si ce beep vous agace).
Le UPPER$ de la ligne 2030 va tout mettre en majuscules. C'est là une sage précaution qui évitera des ratés dans les options Recherches, Sélection ou Tri.
Si vous avez entré une chaîne trop longue, un beep vous prévient qu'il faut la retaper (ligne 2060). Une chaîne vide (ENTER) n'est pas prise en compte (ligne 2070).
En fin de fiche, les valeurs manquantes sont remplacées par un point (lignes 2410-2420).
Vous remarquerez que la création de la page de saisie est un sous-programme à part (lignes 2500 à 2550) parce qu'il sera appelé par d'autres options.
Pour quitter la saisie, descendez le curseur par la touche flèche ''bas'' sur la ligne rouge ''quitter'' ou plus bas qu'elle et ENTER : la dernière fiche tapée est mémorisée et vous revenez au menu principal.

```
2000 ' SAISIE
2010 N=NF+1:GOSUB 2500:H=5
2020 LOCATE 2,23:PRINT "Pour corriger un
e rubrique utiliser les fleches vertical
es."
2030 LOCATE 13,H:LINE INPUT "",R$:R$=UPP
ER$(R$):H=VPOS(#0)-1
2035 IF H=NRU+5 THEN PRINT CHR$(7);
2040 IF H=NRU+6 THEN GOSUB 2400:GOTO 200
0:'FIN DE FICHE
2050 IF H>=NRU+7 THEN GOSUB 2400:GOTO 29
00:'FIN DE SAISIE
2060 IF LEN(R$)>LR(H-5) THEN PRINT CHR$(
7);:GOTO 2030
2070 IF R$<>"" THEN F$(N,H-5)=R$
2080 H=VPOS(#0):GOTO 2030
```

```
2400 ' FIN DE FICHE
2410 FOR R=0 TO NRU:IF LEN(F$(N,R))=0 TH
EN F$(N,R)="."
2420 NEXT:NF=N
2430 RETURN
2500 'ECRAN FICHE
2510 PAPER 2:INK 1,0:PEN 3:CLS
2520 LOCATE 12,2:PRINT "FICHE numero";N:
PEN 1
2530 FOR R=0 TO NRU:LOCATE 10-LEN(F$(0,R
)),5+R:PRINT F$(0,R);" : ";SPC(LR(R));">
"
2540 NEXT
2550 PEN 3:PRINT "  SUIVANTE":PRINT "
QUITTER":PEN 1
2590 RETURN
2900 INK 1,24:RETURN
```

Une remarque concernant la longueur des rubriques et leurs noms : nous sommes en MODE 1 parce que c'est plus lisible, surtout sur l'écran couleur ; donc nous disposons de 40 caractères par ligne à répartir entre le nom, un séparateur, la longueur maxi de la chaîne-rubrique et le repère d'extrémité ''>''.
Notre LINE INPUT démarre en colonne 13, ce qui laisse une longueur maxi de 27 caractères et un nom de rubrique *ne dépassant pas neuf caractères* (ligne 2530). Bien sûr, tout cela est modifiable en changeant notre tabulation. Dans les cas limites, il vous faudra passer en MODE 2 ; pensez en ce cas à modifier le LOCATE de la ligne 2520 : 32 au lieu de 12.

## MODIFICATION ET EFFACEMENT

Lors de l'appel de cette option, on vous demande si vous désirez modifier, effacer ou quitter. Dans les deux premiers cas, on vous demande le numéro de la fiche à traiter. Si vous l'ignorez, revenez à l'option Recherche, que nous verrons plus loin.

```
4000 ' MODIFICATION-EFFACEMENT
4010 INK 1,24:PEN 1:PAPER 0:CLS:LOCATE 7
,12:PRINT "Modifier,Effacer,Quitter ?":T
EX$="MEQ":GOSUB 50000
4020 ON K GOTO 4100,4500,4900
4100 CLS:LOCATE 8,12:INPUT "MODIFIER FIC
HE NUMERO ",N$
```

```
4110 N=VAL(N$):IF N=0 OR N>NF THEN 4100
4120 GOSUB 2500:FOR R=0 TO NRU:LOCATE 13
,R+5:PRINT F$(N,R):NEXT
4130 PEN 3:H=5
4140 LOCATE 13,H:LINE INPUT"",R$:R$=UPPE
R$(R$):H=VPOS(#0)-1
4145 IF H=NRU+5 THEN PRINT CHR$(7);
4150 IF H=NRU+6 THEN N=N+1:IF N<=NF THEN
 GOTO 4120 ELSE GOTO 4000
4160 IF H>=NRU+7 THEN 4000
4170 IF LEN(R$)>LR(H-5) THEN PRINT CHR$(
7);:GOTO 4140
4180 IF R$<>"" THEN F$(N,H-5)=R$
4190 H=VPOS(#0):GOTO 4140
4500 FSEL=0:NS=0:NE=0:E=0
4510 CLS:LOCATE 4,12:INPUT"EFFACER Fiche
 numero (ou Q) ",N$
4520 IF UPPER$(N$)="Q" THEN 4600
4530 N=VAL(N$):IF N=0 OR N>NF THEN PRINT
 CHR$(7):GOTO 4510
4540 E=E+1:S(E)=N:NE=E:GOTO 4510
4600 ' RENUMEROTATION
4610 IF NE=0 THEN 4900
4620 LOCATE 15,15:PRINT "PATIENCE ...":G
OSUB 4700
4630 FOR J=NE TO 1 STEP -1:N=S(J)
4640 FOR T=N TO NF-1
4650 FOR R=0 TO NRU:F$(T,R)=F$(T+1,R):NE
XT
4660 NEXT:NF=NF-1
4670 NEXT:GOTO 4900
4700 'TRI
4710 F=0:S(0)=0:FOR J=1 TO NE
4720 IF S(J)>=S(J-1) THEN 4740
4730 S(0)=S(J):S(J)=S(J-1):S(J-1)=S(0):F
=1
4740 NEXT:IF F=1 THEN 4710
4900 RETURN
```

Aucun risque de demander à modifier une fiche que l'on ne veut pas rectifier (erreur de numéro) : Options ''suivante'' ou ''quitter''.

**La Modification**
C'est la même page d'écran que pour la saisie, mais bien sûr remplie.
La technique d'écriture reste la même : le curseur vous attend au début de la première rubrique et vous pouvez le déplacer par la touche ENTER ou les flèches ''haut'' ou ''bas''.
La grande différence visible est que la chaîne corrigée que vous allez taper va s'écrire en *rouge*, en superposition sur les lettres noires existantes. Ce changement de couleur lève toute ambiguité du genre ''Ai-je corrigé la rubrique précédente ?''.
Pour remplacer une rubrique remplie par ''valeur absente'', il vous faudra taper un point et ENTER ; ne tapez pas des blancs pour effacer l'ancien mot.
Une fois la fiche modifiée, vous retrouverez les mêmes options ''suivante'' ou ''quitter''.

**L'Effacement**
C'est bien sûr plus simple. On entre le ou les numéros de fiches à effacer et en final on tape ''Q''. Apparaît alors le message ''Patience'', car le programme renumérote les fiches suivantes en les ''remontant'' pour combler les vides créés par les fiches effacées.
Le fonctionnement est, lui, assez complexe (lignes 4500 à 4670) :
On va utiliser le tableau DIM S, où on loge seulement les numéros que vous avez entrés. Il y en a NE (=nombre d'effacées).

```
              FICHE numero 210

    TITRE : LE MICRO INSOMNIAQUE >
 CODEDISC : MICRINSO>
    GENRE : AVENTURE     >
  EDITEUR : CAFEINE SOFT    >
   AUTEUR : J.W SCHROUFT    >
     DISC : 12B 14A>
       K7 : .    >
   NOTICE : A IN    >
      REM : TRES DIFFICILE >
  SUIVANTE
   QUITTER



Pour corriger une rubrique utiliser les
fleches verticales.
```

On commence par le bas du ficher, c'est-à-dire que l'on va d'abord traiter (remplacer en remontant) la fiche à effacer de numéro le plus fort. Il est de ce fait prudent d'effectuer un tri préalable sur les numéros de lignes entrés en DIM S(NE) (lignes 4700 à 4740).
On remonte ainsi NE fois les fiches restantes en terminant par l'effacement de la fiche de numéro le plus faible. A chaque effacement le nombre total de fiches NF est diminué de un (ligne 4660).
Attention au piège ! Au vu de votre tableau listé sur imprimante, vous décidez d'effacer les fiches 24, 50 et 78. C'est fait. Ainsi la 25 est devenue la 24 ; la 51 devient la 49 et la 79 devient la 76. OK ? Donc votre listing est faux à partir du n° 24, et il ne faut plus dire en le regardant de nouveau ''Tiens, j'ai oublié d'effacer aussi la 64... ''.
Elle n'est pas celle que vous croyez ! Repassez alors par l'option ''Recherche'' pour en connaître le numéro actuel.

# Chapitre XIII

# RECHERCHE, SELECTION ET TRI DE FICHES

C'est là le grand atout des fichiers informatiques, le côté le plus spectaculaire. Un ''gestionnaire de fichiers'' doit être très performant en ce domaine, sinon il sera rejeté par la ''clientèle'' car la concurrence est très active... L'expérience professionnelle m'a appris que le nouveau-venu à l'informatique est toujours enthousiasmé par ce type de logiciels, mais très rapidement c'est celui-là même qui va devenir le plus exigeant...

L'auteur se trouve alors dans l'obligation de présenter quelque chose qui soit égal ou supérieur aux logiciels du commerce (pour certains, c'est plutôt facile... ). Voilà donc ce qui vous attend :

— Recherche et sélection ne font qu'un. On demande d'abord dans quelle rubrique se situe votre critère ; puis l'élément à y rechercher *ou à rejeter.*
— L'écran vous annonce combien il a trouvé de fiches, et vous demande si vous voulez mieux préciser la sélection (autres critères), ou visualiser la sélection à l'écran, ou l'imprimer ou quitter vers le menu.
— De retour au menu, vous pouvez trier votre sélection en ordre alphabétique par la rubrique de votre choix.
— Imprimer ou sauvegarder votre fichier de sélection ainsi trié.
— Un tri à très grande vitesse (l'auteur en est assez fier...), puisqu'un fichier de 149 fiches à trier par la rubrique TITRE (la plus longue, 21 caractères), ne demande que soixante-trois secondes. Et c'est en Basic.

## RECHERCHE, SELECTION

Nous allons nous servir du tableau DIM S(300), qui va se garnir des *numéros* de fiches sélectionnées. De même que DIM F$ allait de 1 à NF, DIM S va de 1 à NS (NS = nombre de sélectionnées).

L'écran présente la liste des rubriques où va se trouver votre critère de repérage.
Ceci fait, un astérisque rouge se place en face de la rubrique afin de vous rappeler (ou confirmer) votre choix. C'est un petit gadget pour gens étourdis.
En bas d'écran on vous demande alors d'entrer votre élément de repérage :

— Nous utilisons la fonction INSTR ; donc signifiant ''qui contient''. Exemple avec rubrique GENRE vous voulez les jeux d'arcade. Comme élément, vous pouvez entrer ARCADE ou ARCA ou CADE...
— En faisant précéder votre élément du symbole ''flèche en haut'' (à gauche de la touche CLR) signifiera ''ne contenant pas''. (Le choix de ce symbole vient du fait qu'il représente la fonction logique NOT sur les claviers IBM en AZERTY). Un exemple : vous voulez la liste des logiciels dont vous possédez une version sur cassette, genre : ''K7'' : Elément : ''↑.''. C'est-à-dire non manquant, donc existant.
— Si à la question élément vous répondez par la touche ENTER *toutes* les fiches seront sélectionnées. Nous allons voir plus loin que cela peut être utile.
— Vous pouvez taper votre élément en lettres minuscules. Ici comme à la saisie, tout en traité en UPPER$.
— La recherche par la fonction INSTR donne lieu parfois à des surprises : ainsi, si on veut isoler les logiciels se trouvant sur la face B de la disquette n° 4, on a demandé ''DISC'' puis ''4B''. Dans la sélection, on aura aussi les ''14B''... Pas grave, on va demander ensuite ''Préciser''. Encore rubrique DISC, mais on demande différent de 14B, ce qui va éliminer ces indésirables.

La sélection faite (c'est ultra rapide), l'écran affiche, par exemple :

''J'ai trouvé 42 fiches''
''Préciser, Voir, Imprimer, Quitter ?''

— ''Préciser'' va nous représenter la liste des rubriques, élément, etc. Il va de soi que cette deuxième sélection va s'effectuer sur la première sélection, donc moins de fiches à la sortie. On peut demander ''Préciser'' autant de fois que l'on veut, c'est toujours sur la sélection précédente. C'est le flag FSEL qui indique au programme s'il faut travailler sur le fichier global ou sur une sélection (ligne 3100).

```
3000 ' RECHERCHE ET SELECTION DE FICHES
3010 F=0:FSEL=0:PAPER 0:PEN 3:CLS
3020 PAPER 0:PEN 3:CLS
3030 LOCATE 2,2:PRINT "RUBRIQUE pour Rec
herche , Selection :":PEN 1
3040 FOR R=0 TO NRU:LOCATE 15,4+R*2:PRIN
T LEFT$(F$(0,R),1);" - ";F$(0,R):NEXT:LO
CATE 15,22:PRINT "Q - QUITTER"
```

```
3050 PEN 2:TEX$="TCGEADKNRQ":GOSUB 50000
:R=K-1
3060 IF K=10 THEN 3400
3070 LOCATE 1,24:PRINT SPC(39):LOCATE 12
,4+R*2:PEN 3:PRINT "*":NEG=0
3080 PEN 2:LOCATE 2,24:PRINT"QUEL ELEMEN
T (^ pour <>)? ";:PEN 1:INPUT"",EL$:EL$=
UPPER$(EL$):IF LEFT$(EL$,1)="^" THEN NEG
=-1:EL$=RIGHT$(EL$,LEN(EL$)-1)
3085 IF LEN(EL$)>LR(R) THEN PRINT CHR$(7
):GOTO 3070
3090 PEN 1:CLS:LOCATE 15,10:PRINT "PATIE
NCE ..."
3100 IF FSEL=0 THEN 3130
3110 FOR N=1 TO NS:IF (INSTR(F$(S(N),R),
EL$)=0)=NEG THEN F=F+1:S(F)=S(N)
3120 NEXT:NS=F:GOTO 3150
3130 FOR N=1 TO NF:IF (INSTR(F$(N,R),EL$
)=0)=NEG THEN F=F+1:S(F)=N
3140 NEXT:NS=F
3150 LOCATE 9,13:PRINT "J'AI TROUVE";NS;
"FICHES."
3160 LOCATE 4,15:PRINT "Preciser,Voir,Im
primer,Quitter ?":TEX$="PVIQ":GOSUB 5000
0
3170 ON K GOTO 3180,3300,3190,3390
3180 FSEL=1:F=0:GOTO 3020
3190 FSEL=1:GOSUB 6000:CLS:GOTO 3150
3300 ' VISION
3310 FOR I=1 TO NS:N=S(I):GOSUB 2500
3320 FOR R=0 TO NRU:LOCATE 13,5+R:PRINT
F$(N,R):NEXT
3330 H=NRU+6
3340 LOCATE 13,H:INPUT "",Q$:H=VPOS(#0)-
1
3350 IF H<=NRU+6 THEN 3370
3360 IF H>NRU+6 THEN I=NS
3370 NEXT
3380 INK 1,24:GOTO 3000
3390 RETURN
3400 FSEL=0:RETURN
```

— ''Voir'' va montrer à l'écran le contenu des fiches sélectionnées. C'est le même écran qu'en ''Modification'' mais le curseur est déjà sur la ligne ''SUIVANTE'' ; vous n'avez qu'à presser ENTER pour visualiser toutes les fiches. Pour quitter, toujours pareil, descendre d'un cran ou plus par la flèche ''bas'' et ENTER.
— ''Imprimer'' concerne bien sûr votre sélection, et ''Quitter'' renvoie au menu principal.

**NOTE :** Certains ont pu s'étonner de ne pas voir l'option ''VISIONNER'' au menu principal. On peut la remplacer par l'option ''RECHERCHE'' où l'on demande n'importre quelle rubrique et comme élément, on répond par ENTER. Toutes les flèches étant ''sélectionnées'', on demande la sous-option ''VOIR''.
S'il s'agissait de contrôler les dernières fiches saisies vous aurez meilleur compte à demander l'option ''Modification'' puis un numéro de fiche proche de la fin de fichier. Il vous faudra descendre à chaque fois le curseur sur la ligne rouge ''SUIVANTE'' ; c'est un peu moins pratique.
En fait, la pratique montre que lorsqu'un fichier possède déjà quelques dizaines de fiches, il est bien rare que l'on ait envie de les faire défiler une par une à l'écran... C'est donc sans remords que l'on s'est dispensé de créer une option ''Vision''.
Il vous serait facile de l'ajouter en demandant le numéro de départ puis en faisant un GOSUB 3300.

**La fonction NOT**
Matérialisée par NEG = − 1 dans la ligne 3110, elle demande quelques explications de fonctionnement. Petit exercice : Faites PRINT 34 > 22 (en mode direct). L'écran répond − 1, parce que (34 > 22), c'est VRAI. A présent faites PRINT34 > 74, réponse 0, parce que c'est FAUX. Dans le programme on initialise la variable NEG = 0, ligne 3070 (NEG = NEGATIF).
Ligne 3080, si le caractère à gauche de l'élément entré est ''fléché en haut'', NEG est mis à − 1, parce que c'est vrai.
Ligne 3130, la ''grande parenthèse'' contenant INSTR est l'équivalent de ''l'élément n'est pas dans cette fiche''. C'est une affirmation, si elle est vraie, elle vaut − 1, ou zéro si elle est fausse ; on compare donc cette valeur à celle de NEG et si elles sont égales, le numéro de la fiche est sélectionné.
Pour ceux d'entre vous qui découvrent ainsi cette façon de procéder, sachez que cela s'appelle des ''opérations logiques'' (rien à voir avec l'adjectif ''logique'' que vous connaissez). Et si vous voulez ''en mettre plein la vue'' à votre auditoire, utilisez le synonyme plus pompeux de ''opérations BOOLEENNES''...

## LE TRI

Ce serait mentir de dire que c'est simple. L'essentiel est plutôt d'en

retenir la ''stratégie'' générale, et elle au moins est facilement compréhensible.
On devrait plutôt parler de ''classement'', ce serait plus français, mais ''tri'' est entré dans le langage courant informatique.
Il y a deux sortes de tris ; sur des valeurs numériques, c'est du plus petit au plus grand, mais ici nous n'en avons pas, et le tri ''alphanumérique'' (notre cas) qui ne tient compte que des CODES ASCII des caractères rencontrés. On dit alors que ''A''<''B'', d'où l'ordre alphabétique. Après ce petit préambule, revenons au programme.

```
5000 ' TRI
5010 PEN 1:CLS:LOCATE 2,12:PRINT "TRI su
r Fichier,Selection ou Quitter ?":TEX$="
FSQ":PEN 2:GOSUB 50000:CT=K:IF K=3 THEN
5080
5020 CLS:PEN 3:LOCATE 9,2:PRINT "TRI par
 quelle Rubrique ?":PEN 1
5030 FOR R=0 TO NRU:LOCATE 15,4+R*2:PRIN
T LEFT$(F$(0,R),1);" - ";F$(0,R):NEXT
5040 PEN 2:TEX$="TCGEADKNR":GOSUB 50000:
R=K-1:PEN 1
5050 CLS:LOCATE 8,12:PRINT "PATIENCE ...
 TRI EN COURS"
5060 ON CT GOSUB 5100,5700
5070 PRINT CHR$(7)
5080 RETURN
5100 'TRI DU FICHIER
5110 FOR I=1 TO NF:S(I)=I:NEXT:NS=NF
5120 GOSUB 5700
5130 OPENOUT "XXXTRI":PRINT#9,NS
5140 FOR I=1 TO NS:FOR R=0 TO NRU
5150 WRITE#9,F$(S(I),R):NEXT:NEXT
5160 CLOSEOUT
5170 OPENIN "XXXTRI":INPUT#9,NF
5180 FOR N=1 TO NF:FOR R=0 TO NRU:INPUT#
9,F$(N,R):NEXT:NEXT:CLOSEIN
5190 TRI$="XXXTRI.*":|ERA,@TRI$
5200 RETURN
5700 ' TRI SHELL-METZNER
5710 P=NS
5720 P=INT(P/2)
5730 IF P<1 THEN 5800
```

```
5740 DEP=1:FIN=NS-P
5750 D=DEP
5760 C=D+P:IF F$(S(D),R)<=F$(S(C),R) THE
N 5790
5770 S=S(D):S(D)=S(C):S(C)=S
5780 D=D-P:IF D>0 THEN 5760
5790 DEP=DEP+1:IF DEP>FIN THEN 5720 ELSE
 GOTO 5750
5800 RETURN
```

Première question : ''Tri sur le fichier, sur la Sélection, ou Quitter ?'' (Toujours notre issue de secours Q.).
Ensuite, choix de la rubrique sur laquelle va s'effectuer ce tri. Et c'est parti !
Point très important : pendant le tri, le fichier en tableau DIM F$ *ne bouge pas !* C'est le contenu du DIM S qui est trié.
Prenons le cas le plus simple, celui du tri de la sélection.
On explore le tableau S(300) de 1 à NS, qui, rappelons-le encore, ne contient que des numéros de fiches et rien d'autre. Les fiches vont donc être appelées par ces numéros afin de comparaisons, lesquelles vont conduire à des déplacements de valeurs dans le tableau DIM S. En fin de tri, on a dans DIM S le ''bon'' ordre des numéros de fiches. Il est alors très facile d'afficher, d'imprimer ou de sauvegarder cette sélection triée, il suffit d'appeler de 1 à NS les fiches dont les numéros se suivent dans DIM S.
Maintenant, cas du fichier global : on va opérer de même, mais en remplissant artificiellement le DIM S tout bêtement par 1, 2, 3, 4, 5... jusqu'à NF (ligne 5110). Ensuite, le même tri que précédemment dans DIM S. Et c'est très rapide ! Vingt et une secondes pour 149 fiches ! Oui, mais il faut maintenant mettre le fichier global de DIM F$ dans cet ordre. Pas question d'opérer par substitutions, on n'aurait pas assez de mémoire disponible.
Alors astuce n° 2 : on *enregistre* le fichier dans le bon ordre des numéros (lignes 5130-5160) sous le nom sans équivoque de ''XXXTRI''. Tout de suite après, on le charge en mémoire dans DIM F$ (lignes 5170-5180), donc en écrasant l'ancien. Et le tour est joué. Toutefois, si on a utilisé une disquette, on va ensuite y effacer notre ''XXXTRI'' (ligne 5190). Pour 149 fiches, la durée totale TRI + sauvegarde + chargement + effacement = 62 secondes sur disquette.
Avec cassette, il faudra faire un ''REWIND'' après la sauvegarde et ne pas taper la ligne 5190. Ce sera certes plus long que sur disquette ; on a chronométré 4 minutes 32 secondes pour ce fichier de 13 kilo-octets (7 ''blocks''), mais c'est néanmoins bien plus rapide que le tri banal sur le DIM F$ lui-même qui lui nous avait demandé ... 23 minutes.
La méthode de tri utilisée est celle de SHELL-METZNER (lignes 5700-5790). C'est la championne de vitesse, loin devant les autres

méthodes (par tri à bulle, par substitution en escalier, par dichotomie), mais elle présente un léger défaut.
Supposons que vous vouliez trier par Editeurs et en ''second choix'' par titres. Vous seriez tenté de dire ''je vais d'abord trier par titres, et ensuite je referai un tri par éditeurs ; ainsi, dans chaque bloc éditeur, les titres seront dans l'ordre''. Faux avec les tris SHELL-METZNER ou par dichotomie (qui se ressemblent) ; ce serait vrai avec les tris à bulle ou par substitution en escalier, mais ce sont les plus lents... Il fallait bien en choisir un !
Pourquoi la même méthode SHELL-METZNER passe de une à 23 minutes selon que l'on classe des nombres dans DIM S ou des fiches dans DIM F$ ? Parce que dans ce dernier cas, on est obligé de ''déplacer'' une à une les neuf rubriques, à chaque fois que l'on déplace une fiche entière (boucles FOR R=0 TO 8). Dans l'autre cas, c'est seulement un seul nombre entier (sur deux octets). De ce fait, le tri réel du DIM S ne dure que 21 secondes au lieu de 23 minutes pour le DIM F$...

## LA PRATIQUE DES SELECTIONS ET TRIS

Si vous avez une imprimante, il est utile de conserver sur disquette ou cassette plusieurs versions triées du fichier :
Le fichier brut a pour nom LOGICIEL.FIC, c'est celui dans l'ordre chronologique des saisies. Trié par titres, c'est LOGICIEL.TIT ; par genre LOGICIEL.GEN ; par éditeurs LOGICIEL.EDI. Ces trois dernières versions imprimées sont très pratiques.
En revanche, sauvegarder des sélections n'est intéressant que pour ceux qui n'ont pas d'imprimante ; recharger ces petits fichiers est le seul moyen de les consulter.
Un exemple concret : vous voulez la liste de vos jeux d'arcade édités par AMSOFT :

1 — Rechercher ; rubrique : genre, élément = ''arcade''.
2 — Sous-option ''Préciser'' (retour menu précédent).
3 — Recherche : rubrique = éditeur, élément ''Amsoft''.
4 — Sous-option ''quitter'' (retour menu principal).
5 — Option TRI.
6 — Sous-option ''Sélection''.
7 — Tri sur la rubrique ''titre''. Retour menu principal après le beep signalant la fin du tri.
8 — Option impression.
9 — Sous-option ''Sélection''.
10 — Donnez un titre, exemple ''ARCADE de AMSOFT''.

Vous obtenez un tableau par colonnes, une fiche par ligne où les titres sont en ordre alphabétique. Présentation sans bavures.
Il y a ensuite retour au menu principal.
De là, vous pouvez demander une sauvegarde de ce petit fichier sous le nom de AMSOFT.ARC.

Après quoi vous pouvez poursuivre l'exploitation de votre fichier global, faire une autre sélection, etc.

## CONCLUSION

Vous avez remarqué que l'intérêt de notre logiciel sur la plupart de ses concurrents du commerce est qu'il peut conserver en mémoire deux fichiers, le global est une sélection ; ce dernier tenant au grand maximum 600 octets en mémoire.
De ce fait, toutes les options du menu principal peuvent s'appliquer à l'un ou à l'autre.
La manipulation étant simple (parce que claire) et souple, on peut se livrer sans risques à d'innombrables triturations de fiches.

## LES METHODES DE TRI

Nous n'en citerons que deux, la plus simple (et aussi la plus lente) et la plus rapide. Un petit banc d'essai comparatif va les départager nettement.

```
10 'TRIDEMO - DEMONSTRATION DE TRIS
20 DEFINT A-S,U-Z
50 DIM X(30),V(30)
60 FOR I=1 TO 30:X(I)=INT(RND*100):V(I)=
X(I):NEXT
70 CLS:PRINT "FICHIER BRUT :":PRINT
80 FOR I=1 TO 30:PRINT V(I);:NEXT:PRINT:
PRINT
100 ' TRI A BULLE
110 T0=TIME
120 ' FD=FLAG DEPLACEMENT:B=BULLE
130 FD=0:B=0:FOR I=1 TO 30
140 IF V(I)>=V(I-1) THEN 160
150 B=V(I):V(I)=V(I-1):V(I-1)=B:FD=1
160 NEXT
170 IF FD=1 THEN 130
180 T1=TIME
190 PRINT "TRI A BULLE :";(T1-T0)/300;"s
":PRINT
200 FOR I=1 TO 30:PRINT V(I);:NEXT:PRINT
:PRINT
300 ' TRI SHELL-METZNER
```

```
310 T0=TIME
320 M=30 :'MAXI DU FICHIER
330 M=INT(M/2) :' MILIEU
340 IF M<1 THEN 410
350 DEPART=1:FIN=30-M
360 D=DEPART
370 I=D+M:IF X(D)<=X(I) THEN 400
380 B=X(D):X(D)=X(I):X(I)=B
390 D=D-M:IF D>0 THEN 370
400 DEPART=DEPART+1:IF DEPART>FIN THEN 3
30 ELSE GOTO 360
410 T1=TIME
420 PRINT "TRI SHELL-METZNER EN";(T1-T0)
/300;"s":PRINT
430 FOR I=1 TO 30:PRINT X(I);:NEXT:PRINT
440 ' --------- FIN DE LISTING ---------
```

Nous remplissons deux tableaux identiques DIM X et DIM V de 30 nombres aléatoires de 0 à 99 (ligne 60). Puis chacun sera trié par l'une de ces méthodes avec chronométrage par la fonction TIME.
Le tri à bulle a une durée moyenne de cinq secondes contre une seconde et demi pour le tri Shell-Metzner. Ce rapport de temps serait bien plus grand si le nombre d'éléments était très supérieur à trente.
Le principe du tri à bulle (ligne 100 à 200) est ultra simple : si deux éléments consécutifs ne sont pas dans l'ordre, on les inverse et on met un flag (FD) à 1. Puis on ré-explore le fichier de haut en bas autant de fois que l'on retourne un flag égal à 1. Avantages :

— si le fichier est déjà classé, un seul passage suffit ; c'est une vérification ultra-rapide ;
— son écriture reste très simple : cinq lignes Basic ;
— dans le cas d'un tri sur un DIM à plusieurs rubriques, on peut faire des tris successifs en commençant par le moins prioritaire : l'ancien ordre sera conservé si le tri suivant tombe sur des égalités (le Shell-Metzner mélange tout).

On utilisera donc le tri à bulle pour des petites opérations de tris (moins de 20 éléments).
Le tri Shell-Metzner (lignes 300 à 430) est d'un fonctionnement plus complexe. Il divise chaque fois le fichier en deux et compare des éléments dans chacune des moitiés. Cela diminue le nombre de déplacements, d'où un gain de temps. Pour vérifier un fichier déjà trié, il mettra un peu plus de temps que le tri à bulle. On le réserve pour les tris sur des nombreux éléments.

# Chapitre XIV

# LES ENTRÉES/SORTIES DE FICHIERS

Cela sous-entend les opérations de *chargement, sauvegarde et édition* sur imprimante. Si les deux premières sont simples et banales à concevoir, il en va autrement en ce qui concerne les programmes d'édition, du moins quand on veut faire quelque chose de ''propre'' sur le plan esthétique.

## LA SAUVEGARDE (lignes 7000 à 7900)

On demande s'il s'agit du fichier ou de la sélection, et toujours le ''quitter'' de secours. Ensuite, le nom que portera cet enregistrement. Là, un petit gadget pour fainéants : en répondant par ENTER, le nom sera LOGICIEL.LOG, celui du fichier brut, celui que l'on ré-enregistre après chaque séance de saisie.
La syntaxe de ce nom est contrôlée par un GOSUB 8500. C'est facultatif pour les utilisateurs de cassettes, mais c'est néanmoins vivement conseillé car le jour où vous aurez un drive, vous transférerez vos programmes de cassettes à disquettes sans avoir la corvée de modifier les noms. Rappelons ces règles de syntaxe : nom de huit caractères maxi sans espaces ni ponctuations puis, c'est facultatif, un point et un ''nom d'extension'' de trois caractères maxi. Les noms d'extension ''réservés'' (interdits) sont BAS, BAK, BIN, COM et EXE.
Par défaut de nom d'extension, le programme ajoute ''.LOG''. En effet, il est imprudent d'enregistrer sur disquette des fichiers ASCII sans noms d'extension, car plus tard on ne sait plus quel programme a généré ce fichier lu en faisant CAT. En revanche, plus d'ambuiguité si l'extension est l'abréviation de ce logiciel ; exemple : .LOG qui vient du nom de ce programme LOGICLAS.BAS.
Tout d'abord, nous enregistrons une valeur numérique NF ou NS, puis des valeurs alphanumériques (ou chaînes), le contenu de chaque fiche. Pour 100 fiches, il y a donc 901 enregistrements. Heureusement qu'AMSTRAD est rapide...

### LE CHARGEMENT D'UN FICHIER (lignes 8000 à 8570)

Nous retrouvons la même structure que précédemment à la différence que l'on ne demande plus "Fichier ou Sélection ?", c'est toujours Fichier, même s'il s'agit de charger en DIM F$ un enregistrement d'une sélection antérieure. OK ?

```
7000 ' SAUVEGARDE
7010 CLS:LOCATE 1,12:PRINT"SAUVEGARDE du
 Fichier,Selection,Quitter?":TEX$="FSQ":
GOSUB 50000
7020 ON K GOTO 7030,7100,7900
7030 CLS:LOCATE 2,15:PRINT "(pour le Fic
hier LOGICIEL.LOG = ENTER)"
7040 LOCATE 5,10:INPUT"SAUVER le Fichier
 ",FICH$
7050 GOSUB 8500:IF REFUS THEN 7030
7060 LOCATE 12,18:PRINT "PATIENCE ..."
7070 OPENOUT FICH$:PRINT#9,NF
7080 FOR N=1 TO NF:FOR R=0 TO NRU
7090 WRITE#9,F$(N,R):NEXT:NEXT:CLOSEOUT:
GOTO 7900
7100 CLS:LOCATE 2,10:INPUT"NOM de la Sel
ection ",FICH$:LOCATE 12,18:PRINT "PATIE
NCE ..."
7110 GOSUB 8500:IF REFUS THEN 7100
7120 OPENOUT FICH$:PRINT#9,NS
7130 FOR I=1 TO NS:N=S(I):FOR R=0 TO NRU
7140 WRITE#9,F$(N,R):NEXT:NEXT:CLOSEOUT
7900 RETURN
8000 ' CHARGEMENT
8010 CLS:LOCATE 2,15:PRINT "(pour le Fic
hier LOGICIEL.LOG = ENTER)"
8020 LOCATE 5,10:INPUT"Charger le Fichie
r ",FICH$
8030 IF FICH$="" THEN FICH$="LOGICIEL.LO
G":GOTO 8050
8040 GOSUB 8500:IF REFUS THEN 8000
8050 LOCATE 12,18:PRINT "PATIENCE ..."
8060 OPENIN FICH$:INPUT#9,NF
8070 FOR N=1 TO NF:FOR R=0 TO NRU:INPUT#
9,F$(N,R):NEXT:NEXT:CLOSEIN
```

```
8080 RETURN
8500 ' NOM DE FICHIER
8510 REFUS=0:IF FICH$="" THEN FICH$="LOG
ICIEL.LOG":GOTO 8570
8520 EXT=INSTR(FICH$,"."):LNF=LEN(FICH$)
8530 IF EXT>9 OR INSTR(FICH$," ") THEN R
EFUS=1
8540 IF EXT=0 AND LNF>8 THEN REFUS=1
8550 IF REFUS THEN PRINT CHR$(7);:GOTO 8
570
8560 IF EXT=0 THEN FICH$=FICH$+".LOG"
8570 RETURN
```

Il y a en plus le sous-module contrôle du nom de fichier (ligne 8500-8570) utilisé aussi pour la sauvegarde.

**L'IMPRESSION**
**(lignes 6000-6330)**

L'édition d'un fichier sur imprimante doit répondre aux exigences suivantes :

- L'exploitation de l'édition papier doit être pratique, c'est-à-dire pouvoir localiser rapidement sur le papier ce que l'on y cherche.
- Une fiche par ligne.
- Le contenu des rubriques parfaitement aligné en colonnes.
- Les colonnes sont suffisamment espacées ou séparées par un trait vertical.
- Un trait vertical ne doit pas ''toucher'' le texte.
- Les colonnes sont légendées en haut de page, et ces légendes sont ''isolées'' par un trait horizontal.
- Chaque page ne comporte que 50 fiches ; ceci afin de rendre instantanée la recherche par numéro de fiche (exemple, n° 132 = page 3).
- Le saut de page est automatique, avec numérotation de la page et répétition des légendes de colonnes. Bien sûr, sans décalage vertical.
- Tout ce qui est imprimé sur une page doit être présenté dans un encadrement rectangulaire. C'est une assurance de non-coupures en cas de photocopie.
- Les nombres d'une colonne de valeurs numériques doivent être imprimés en PRINT USING.
- Les valeur manquantes doivent être remplacées par un ou des points (c'est déjà fait) ; afin que le lecteur puisse apprécier l'alignement sans devoir utiliser une règle...

Quel cahier des charges démentiel ! Mais non, cela ne fait que 29 petites lignes de Basic pour une présentation classe ''PRO''. J'oubliais encore une condition : un titre pour le tablau est souhaitable ; il peut ne pas être répété à chaque page ; en tout cas il doit être automatiquement centré sur sa ligne.
A présent, abordons notre listing :

**La structure de page**
Elle tient compte que l'on utilise du papier mesurant onze pouces (27,9 cm) entre chaque pliure ; c'est la dimension la plus courante, soit 11 × 6 = 66 lignes par page.
Nous aurons donc l'ordre d'un trait horizontal, suivi du numéro de page, les légendes de colonnes, autre trait horizontal, 50 lignes-fiches et un troisième trait horizontal final. Soit 3 + 50 + 1 = 54 lignes à imprimer ; il y aura donc un saut de page de 66 × 54 = 12 lignes ''blanches'' à effectuer (ligne 6320).

```
6000 ' EDITION
6010 CLS:LOCATE 1,12:PRINT"IMPRESSION du
 Fichier,Selection,Quitter?":TEX$="FSQ":
GOSUB 50000
6020 IF K=3 THEN 6190
6030 WIDTH 137:PRINT #8,CHR$(27);CHR$(64
);:P=1
6040 CLS: LOCATE 10,12:LINE INPUT"TITRE
: ",TIT$:IF TIT$="" THEN TIT$="  "
6050 IF K=1 THEN LOCATE 6,14:INPUT"A PAR
TIR DE QUEL NUMERO ? ",ND$:ND=VAL(ND$):I
F ND=0 OR ND>NF THEN 6050
6060 LIG=7:FOR R=0 TO NRU:LIG=LIG+LR(R)+
2:NEXT
6070 PRINT#8,SPC((78-LEN(TIT$))/2);TIT$:
PRINT#8,CHR$(15)
6080 TRAIT$=STRING$(LIG,"-")
6090 PRINT#8,TRAIT$;P
6100 N=0:GOSUB 6200:PRINT#8,TRAIT$:L=5
6110 IF K=2 THEN 6150
6120 FOR N=ND TO NF:GOSUB 6200:L=L+1:IF
L=55 THEN GOSUB 6300
6130 NEXT
6140 GOTO 6170
6150 FOR I=1 TO NS:N=S(I):GOSUB 6200:L=L
+1:IF L=55 THEN GOSUB 6300
```

```
6160 NEXT
6170 PRINT#8,TRAIT$
6180 PRINT#8,CHR$(27);CHR$(64)
6190 RETURN
6200 ' LIGNE FICHE
6210 PRINT#8,"|";:PRINT#8,USING"####";N;
6220 FOR R=0 TO NRU:PRINT#8,"  ";F$(N,R)
;SPC(LR(R)-LEN(F$(N,R)));:NEXT:PRINT#8,"
 |"
6230 RETURN
6300 ' SAUT DE PAGE
6310 PRINT#8,TRAIT$:P=P+1:AL=N:N=0
6320 FOR J=1 TO 12:PRINT#8:NEXT:PRINT#8,
TRAIT$;P:L=5
6330 GOSUB 6200:PRINT #8,TRAIT$:L=5:N=AL
6340 RETURN
```

Deuxièmement il faut calculer la longueur horizontale LIG de chaque ligne (ligne 6060). Nous allons prévoir dans l'ordre gauche à droite : Une barre verticale, un blanc, trois cases pour le numéro de fiche et deux blancs de séparation. Total = 7 cases ; puis chaque rubrique prolongée par deux blancs de séparation. En fait, le dernier blanc sera une barre verticale.
Définissons le trait horizontal TRAIT$ comme une suite de tirets de longueur LIG (ligne 6080).
Etablissons le sous-programme d'impression d'une ligne-fiche (lignes (6200-6230) : vous remarquerez que pour chaque rubrique, on écrit à la suite le nombre de blancs pour compléter. C'est un peu lourd mais beaucoup plus sûr que les commandes de tabulations sur imprimante.
Puis le sous-programme saut de page (6300-6340). On y trouve le TRAIT$ de bas de page, le saut de douze lignes, le comptage des pages P, la mémorisation de la dernière ligne imprimée AN ( = ancien N), afin de pouvoir ré-éditer la ligne zéro du DIM F$ (contient les légendes), pour la restituer ensuite par N = AL (ligne 6330). Le L = 5 des lignes 6330 et 6100 est une initialisation du comptage des lignes imprimées.
Voilà ! Nous venons de faire les 80 % du travail, la suite sera banale. Vous remarquerez que pour adapter ce module à un autre programme, il n'y a rien à retoucher.

**Le déroulement de l'impression**
Il demande tout d'abord si vous désirez imprimer le fichier ou votre sélection. Premier cas, on vous demande le numéro de départ ND ; second cas, on utilisera le passage 6150.
Dans les deux cas, on vous propose d'entrer un titre pour votre édition. Ce titre sera imprimé en caractères normaux (dits PICA), auto-

centré (ligne 6070), puis passage en écriture dite condensée, code CHR$(15), 137 caractères par ligne. Toute édition doit commencer et finir par une vidange du buffer de l'imprimante. C'est le CHR$(27); CHR$(64); des lignes 6030 et 6180.
Résumons : On positionne le papier sur l'imprimante. Impression du titre. Saut d'une ligne blanche. Trait, légendes, trait, 50 fiches, trait et saut de page.

**NOTE** : Le titre n'apparaît que sur la page 1, mais la ligne légendes occupe toujours la même position en hauteur sur chaque page.

On pourrait encore peaufiner la présentation en ce qui concerne les traits horizontaux et verticaux, mais ça c'est du ''troisième degré'' que nous aborderons au chapitre XVII.

## LES MODULES ANNEXES (50000 et 60000)

Rien à taper, ce sont nos routines utilitaires que l'on ajoute au programme par MERGE. Pour le module 60000, c'est notre ''DISCUT'' décrit au chapitre III mais renuméroté. Pour ce dernier, nous avons procédé de la manière suivante :

- LOAD ''DISCUT''
- On supprime les lignes qui se rapportent à l'option ICPM parce que trop dangereuses dans ce programme (perte du Basic !).
- RENUM 60000
- SAVE ''DIS'' (ou SAVE ''DIS '',A pour les CPC 464)
- MERGE ''DIS''

Cette option ''DISC-UTIL'' est dans la pratique très souvent appelée : pour s'assurer de l'orthographe d'un nom de fichier au CATalogue, ou pour s'assurer qu'il y est bien..., pour voir s'il reste de la place, pour effacer les ''.BAK'', pour changer le nom d'un fichier, etc... et tout cela en partant du menu principal et en y revenant sans rien perdre.
Si vous n'avez pas encore de DRIVE, tapez quand même cette option, elle ne vous gênera pas pour l'usage de cassettes.

```
50000 'REPONSE A UN MENU
50010 LT=LEN(TEX$)
50020 LOCATE 15-LT,24:PRINT"Reponse (";
50030 FOR I=1 TO LT-1
50040 PRINT MID$(TEX$,I,1);",";:NEXT
50050 PRINT RIGHT$(TEX$,1);")";CHR$(154)
;CHR$(243);CHR$(207)
50060 TEX$=UPPER$(TEX$)
```

```
50070 R$="":WHILE R$="":R$=INKEY$:WEND
50080 R$=UPPER$(R$):K=INSTR(TEX$,R$)
50090 IF K=0 THEN PRINT CHR$(7);:GOTO 50
070
50100 RETURN
60000 '*** DISCUT *** Utilitaire pour DI
SC
60010 CLS
60020 PRINT "Cat  Erase  Rename  Drive
Bak- Quitter"
60030 Q$="":WHILE Q$="":Q$=UPPER$(INKEY$
):WEND
60040 Q=INSTR("CERDBQ",Q$):IF Q=0 THEN P
RINT CHR$(7);:GOTO 60030
60050 IF Q=6 THEN RETURN
60060 ON Q GOSUB 60080,60090,60130,60190
,60240
60070 GOTO 60020
60080 CAT:RETURN
60090 INPUT "  ERASER quel Fichier ? ",F
$:GOSUB 60500:IF REFUS THEN 60120
60100 IF F$="*.*" THEN PRINT"  Etes-vous
 bien sur ?...(O/N) ";:R$="":WHILE R$=""
:R$=UPPER$(INKEY$):WEND:IF R$="N" THEN 6
0120 ELSE IF R$<>"O" THEN 60100
60110 |ERA,@F$
60120 RETURN
60130 INPUT "  ANCIEN NOM : ",F$:IF INST
R(F$,"*") >0 THEN 60130
60140 GOSUB 60500:IF REFUS THEN 60180 EL
SE A$=F$
60150 INPUT " NOUVEAU NOM : ",F$:IF INST
R(F$,"*") >0 THEN 60150
60160 GOSUB 60500:IF REFUS THEN 60180 EL
SE N$=F$
60170 |REN,@N$,@A$
60180 RETURN
60190 PRINT "  SUR QUEL DRIVE (A/B) ?"
60200 D$="":WHILE D$="":D$=UPPER$(INKEY$
):WEND
60210 IF D$<>"A" AND D$<>"B" THEN 60200
60220 |DRIVE,@D$
```

```
60230 RETURN
60240 F$="*.BAK"
60250 |ERA,@F$
60260 RETURN
60500 'SECURITE DU NOM DE FICHIER
60510 L=LEN(F$):REFUS=0
60520 IF INSTR(F$,CHR$(32)) >0 OR L=0 TH
EN REFUS=1
60530 P=INSTR(F$,".")
60540 IF P>9 OR (P=0 AND L>8) THEN REFUS
=1
60550 IF P>0 AND L-P>3 THEN REFUS=1
60560 IF REFUS THEN PRINT CHR$(7);
60570 RETURN
65535 ' ------- FIN DE LISTING -------
```

LOGICIELS en CASSETTES

| 0 | TITRE | CODEDISC | GENRE | EDITEUR | AUTEUR | DISC | K7 | NOTICE | REM |
|---|---|---|---|---|---|---|---|---|---|
| 1 | AMLETTRE | AMLETTRE | TRAIT. TEXTE | AMSOFT | . | 1A+6A | ORIGI | F | CPC 1 |
| 2 | CPC464 | CPC464 | DEMO CPC | AMSTRAD | . | 11B | ORIGI | . | . |
| 3 | DECATHLON | . | SPORT | OCEAN | DALEY THOMPSON | . | ORIGI | A | . |
| 4 | FIGHTER PILOT | FP | SIM. VOL | DIGITAL INTEGRA | MARSHALL | 4A | ORIGI | . | . |
| 5 | GESTION DE FICHIERS | GESTFICH | FICHIER | CORE | ABAD | 6A BA | ORIGI | F | CPC 1 |
| 6 | RALLY II | RALLY2 | AUTO | LORICIEL | CARLO PERCONTI | 12P | ORIGI | F | . |
| 7 | WATSON | WATSON | ASSEMBLEUR | MICRO APPLICATI | DR WATSON | 9A | ORIGI | LIVRE | +UTILITAIRES |

# Chapitre XV

# LA RÉCUPÉRATION DE FICHIERS

Il s'agit de fichiers antérieurs que vous aviez enregistrés à l'aide de logiciels du commerce (gestionnaire de fichiers, traitements de texte, etc.) et que vous aimeriez bien récupérer en les mettant *sous une autre forme* (ex.: tableaux DIM), afin de pouvoir les utiliser avec un programme de votre fabrication.
En faisant type nom du fichier sous CPM, vous retrouvez votre texte mais entrecoupé de caractères graphiques bizarres qui sont pour le logiciel des ''caractères de contrôle'' (signaux de commande). Sachant pertinemment ce que vous avez entré, vous aurez vite fait de trouver le rôle de chacun d'eux : donc ceux que vous ferez sauter, et ceux que vous transformerez en ordres Basic classiques.
Attention ! On ne tombe pas sur le coup d'une loi : on ne déplombe pas un logiciel commercial, on décode un *fichier personnel* obtenu par un logiciel. Nuance... Certains éditeurs proposent tel logiciel A pouvant utiliser les fichiers écrits par leur logiciel B : c'est là un avantage optionnel non une obligation, sinon ce serait de la ''vente forcée''. Voilà pourquoi l'auteur a tout à fait la conscience tranquille en publiant ce qui suit.

## DECODAGE D'UN FICHIER ASCII

Ceux qui n'ont pas de DRIVE n'ont pas le CPM avec TYPE ou ED. Aucune importance, puisque TYPE et ED sont impuissants face à certains ''pièges à loups'', d'où clignotement de l'écran et plantage complet de la machine ! Un petit exemple : si on rencontre CHR$(12), c'est l'effacement de l'écran ; sur imprimante, c'est l'ordre d'avancer le papier d'une page ! Très gai.
En conséquence, nous avons conçu le petit programme ASCII nommé DECODFIC auquel rien ne résiste (convient aussi aux cassettes).

```
10 '- DECODFIC - EXTRAIT LES CARACTERES
AFFICHABLES D'UN FICHIER COMMERCIAL
20 'AMSTRAD CPC - M.ARCHAMBAULT - 4/86
30 CLS:D=0:INPUT "IMPRIMANTE ? (O/N) ",Q
$:IF UPPER$(Q$)="O" THEN D=8
40 CLS:INPUT"NOM DU FICHIER: ",FICH$
50 OPENIN FICH$:F=0
60 WHILE NOT EOF:F=F+1
70 PRINT#D,F;
80 LINE INPUT#9,A$
90 FOR I=1 TO LEN(A$)
100 C$=MID$(A$,I,1)
110 IF ASC(C$)>31 AND ASC(C$)<127 THEN P
RINT#D,C$;:ELSE PRINT#D,"<";ASC(C$);">";
120 NEXT:PRINT#D,CHR$(13)
130 WEND
140 CLOSEIN
150 ' ---------- FIN DE LISTING --------
```

Très simple : chaque caractère de l'enregistrement A$ reçu est inspecté. Si son code ASCII est compris entre 32 et 126, on l'affiche, sinon c'est son code ASCII disposé entre ''< >''. De plus, chaque enregistrement est numéroté. Essayez-le avec un fichier obtenu avec un traitement de texte (''AMLETTRE'', ''TEXTOMAT'', etc.). Amusant et instructif. Pour le tester à fond, nous avons créé un fichier très ''méchant'' par :

```
10 OPENOUT''PIEGES''
20 FOR I=0 TO 31:WRITE#9,CHR$(I):NEXT
30 CLOSEOUT
```

Si vous aimez les feux d'artifice, essayez, sous CPM, TYPE PIEGES. Traité par DECODFIC, tout se passe bien sauf pour 13 et 26 qui provoquent un saut de ligne, donc un saut de +1 dans la numérotation. Normal pour CHR$(13) = retour de ligne ; et CHR$(26), c'est le CTRL Z signifiant fin de fichier. Il n'empêche que les codes 27 à 31 sont listés. Alors qu'essayé sous CPM par ED PIEGES, tout se passe bien jusqu'à 25, après fin du fichier...
Le second travail va être de concevoir un programme Basic qui va lire le fameux fichier, enlever ou traduire tous ces codes de contrôle dont on n'a plus que faire, mettre en tableau DIM les chaînes récupérées et ''mises au propre'' et finalement sauvegarder ces données normalement par un OPENOUT. Ce travail prendra moins de temps que si vous deviez re-entrer au clavier tout ce que vous aviez déjà tapé avec le logiciel d'origine...

## LE PROGRAMME DE RECUPERATION

Nous prendrons un exemple concret qui se rapporte au gestionnaire de fichiers sans doute le plus répandu, celui de CORE. Célèbre parce que simple d'emploi, bon marché, en français, avec un excellent rapport prestation/prix. Rares sont les ''Amstradistes'' qui ne l'ont pas utilisé.
Si vous avez tapé notre gestionnaire LOGICLAS des chapitres précédents, ou une adaptation, vous aimeriez bien récupérer le gros fichier obtenu par CORE afin de pouvoir l'exploiter par un logiciel nettement plus performant.
Le programme qui suit, baptisé ''CORASCII'' charge n'importe quel fichier obtenu par CORE, affiche à l'écran, ou mieux sur imprimante, le nombre de fiches, nom et nombre des rubriques, la plus grande longueur de chaîne rencontrée dans chaque rubrique, puis le total que cela représente. Vous avez alors tous les éléments pour définir votre futur gestionnaire de fichiers. En final, la sauvegarde en ASCII des fiches au ''format'' utilisé par LOGICLAS, à savoir nombre NF puis les fiches. Nous avons ainsi récupéré en quelques dizaines de secondes deux fichiers de 14 kilo-octets... C'est rentable...

Le codage du fichier CORE est le suivant :
— nom du fichier
— 20 lignes de noms de rubriques
— nombre de fiches +2

L'enregistrement n° 23 est donc la première fiche. En effet, toutes les rubriques d'une fiche sont concaténées avec des repères de séparation. Chaque enregistrement-fiche commence par CHR$(249), le chiffre 1 (=numéro de rubrique), un blanc, le contenu de la rubrique n° 1. Les séparateurs inter-rubriques sont différents : CHR$(154);CHR$(249), n° de rubrique, un blanc, le contenu. La fin de fiche est repérée par CHR$(164);CHR$(181);CHR$(164). Une donnée ''hors rubrique'' est signalée par un CHR$(164) tout seul.
Pas simple ! Il va falloir hâcher menu à coups de INSTR et de MID$ ! Il faudra aussi éliminer ces données hors rubriques puisqu'étant non logeables dans un tableau DIM.

Le début de l'analyse est facile : on met en DIM RUB$ les noms de rubrique (lignes 100 à 120), on récupère le nombre de fiches NF, ce qui nous permet alors de fixer notre tableau DIM F$ (lignes 130, 140). En ligne zéro, on loge les noms de rubrique et on ERASE le DIM RUB$ dont on n'a plus besoin.
C'est alors la dissection des lignes-fiches, une par une. Cela consiste à récupérer le numéro de rubrique succédant à un ''séparateur'' puis la chaîne entre le blanc qui le suit et le séparateur suivant (lignes 200 à 430). Puis la chaîne ainsi isolée est enfin logée dans notre tableau DIM F$ (ligne 290).
Pour effectuer tout ce travail sur un fichier CORE de 14 kilo-octets, nous avons chronométré vingt-six secondes !

Cette analyse-traitement de fichier "commercial" doit être ici *considérée comme un exemple* qui montre que, dès que l'on a découvert tout le système de codage d'un fichier, il n'est pas difficile de le récupérer en DIM.
La suite, à savoir la sauvegarde, l'affichage des résultats (lignes 600 à 710), la recherche de la plus grande longueur dans chaque rubrique (lignes 800 à 860), sont banales. Et surtout *réutilisables* pour un fichier venant d'un autre logiciel. Vous n'aurez à reprogrammer que la partie lecture + analyse (lignes 100 à 430).

```
10 ' CORASCII - Fichier "GESTION DE FICH
IERS" de "CORE"--> Fichier ASCII
20 'AMSTRAD CPC-Michel Archambault 1986
30 OPENOUT "BIDON":MEMORY HIMEM-1:CLOSEO
UT
40 DEFINT A-Z
50 DIM RUB$(20)
60 S$=CHR$(164)+CHR$(249):'SEPARATEUR CO
RE
70 CLS:LOCATE 3,10:INPUT "Nom du Fichier
 CORE : ",FICH$
80 OPENIN FICH$
90 INPUT#9,A$:I=0:'INDICE RUBRIQUE
100 FOR N=2 TO 21:INPUT#9,A$
110 IF A$<>"" THEN RUB$(I)=A$:I=I+1
120 NEXT
130 INPUT#9,NB:NF=NB-2
140 DIM F$(NF,I)
150 FOR J=0 TO I:F$(0,J)=RUB$(J)
160 NEXT:ERASE RUB$
200 'DECOUPE DES ENREGISTREMENTS
210 FOR N=1 TO NF:LINE INPUT#9,A$:L=LEN(
A$)
220 P=INSTR(A$,S$)
230 PR=VAL(MID$(A$,3,1)):'PREMIERE RUBRI
QUE
240 IF PR>0 THEN F$(N,PR-1)=MID$(A$,5,P-
5)
250 Q=INSTR(P+1,A$,S$)
260 C=VAL(MID$(A$,P+2,2)):Z=4:IF C>9 THE
N Z=5
```

```
270 IF Q=0 THEN 320
280 IF C=0 THEN 300 ELSE C=C-1
290 F$(N,C)=MID$(A$,P+Z,Q-P-Z)
300 P=Q
310 GOTO 250
320 IF C>0 THEN F$(N,C-1)=MID$(A$,P+Z,L-
3-P-Z)
400 'ELIMINATION DES DONNEES HORS RUBRIQ
UES
410 FOR J=0 TO I-1:S=INSTR(F$(N,J),CHR$(
164))
420 IF S THEN F$(N,J)=LEFT$(F$(N,J),S-1)
430 NEXT:NEXT:CLOSEIN
500 'SAUVEGARDE
510 CLS:LOCATE 15,5:PRINT "SAUVEGARDE :"
:LOCATE 8,10:INPUT "NOM DU FICHIER : ",F
IC$
520 OPENOUT FIC$
530 PRINT #9,NF
540 FOR N=1 TO NF:FOR J=0 TO I-1
550 WRITE#9,F$(N,J):NEXT:NEXT:CLOSEOUT
600 'LEGENDE
610 CLS:W=0
620 TOT=0:LOCATE 1,1:PRINT#W, "CARACTERI
ST. DU FICHIER ";:PEN 3:PRINT#W,FIC$;:PE
N 1:PRINT#W," :"
630 LOCATE 6,3:PRINT#W,NF;"fiches avec";
I;"rubriques :"
640 PRINT#W :FOR J=0 TO I-1
650 PRINT#W,USING"######";J;:PRINT#W,"-"
;F$(0,J);:GOSUB 800
660 NEXT
670 PRINT#W:PRINT#W,TAB(10);"Total=";TOT
;"caracteres":PRINT#W
680 LOCATE 12,25:PRINT  "IMPRIMER ? (O/N
 ) ":Q$="":WHILE Q$="":Q$=INKEY$:WEND
690 IF UPPER$(Q$)="O" THEN W=8 :GOTO 620
:ELSE W=0
700 IF UPPER$(Q$)<>"N" THEN 680
710 CLS:END
800 'LONGUEURS MAXI
810 M=1:FOR N=1 TO NF
```

```
820 IF LEN(F$(N,J)) >M THEN M=LEN(F$(N,J
))
830 NEXT
840 PRINT#W,TAB(23);USING"###";M;:PRINT#
W," caract."
850 TOT=TOT+M
860 RETURN
870 ' -------- FIN DE LISTING --------
```

# Chapitre XVI

# LA COPIE D'ÉCRAN SUR IMPRIMANTE

C'est ce que l'on appelle aussi le ''Hard Copy''. Sur AMSTRAD, la chose est plus compliquée que sur les autres micros car le plan de la mémoire d'écran est d'une extrême complexité, ce qui explique la relative lenteur de l'affichage de texte. Le fabricant n'a pas compliqué par plaisir, c'est la rançon de l'écran unique (texte et graphisme), qui permet aussi les trois MODE 0, 1 et 2 (pour plus de détails sur la mémoire d'écran, voir MPSA chapitre XV).
Côté imprimante, il faut le ''BIT IMAGE'', c'est-à-dire ESC K ou L ou M, ce que possèdent pratiquement tous les modèles au ''Standard EPSON'' ; ce qui exclut la DMP 1. De quoi s'agit-il ?
L'imprimante ne reçoit de l'ordinateur que des nombres de 0 à 255. En mode normal, elle les traduit par un dessin de caractères, exemple 65 donne le A (=code ASCII). La tête d'impression comporte neuf aiguilles superposées, actionnées par des micro-électro-aimants. En fait, ce sont les 7 du haut ou les 7 du bas qui peuvent fonctionner simultanément : ces derniers sont rarement sollicités, pour les minuscles avec jambage inférieur : g, j, p, q, y, ç et la virgule.
Les alignements verticaux de points sont des ''images binaires'' de nombres de 0 à 127 fournis par la ROM de l'imprimante. Généralement une salve de cinq nombres pour dessiner un caractère.

En mode ''bit image'', l'imprimante interprète le nombre reçu non plus comme un code ASCII mais pour en reproduire l'image binaire, de haut en bas. Ainsi, 67 donne 1000011, c'est-à-dire un petit point en haut et deux en bas.
Sur l'image vidéo, c'est le même principe mais les images binaires sont dans le sens horizontal... Imaginez une image d'écran en MODE 2, caractères noirs sur fond blanc : il y a 200 lignes comportant chacune 80 images binaires d'octets ; chaque point noir (PEN 1), c'est un 1, un point blanc (PAPER 0), c'est un zéro.

Nous allons explorer 200 × 80 × 80 = 128 000 points sur l'écran à l'aide de la fonction BASIC TEST, traduire ces mesures en images binaires verticales que nous enverrons vers l'imprimante.
Sur ce principe, nous vous présentons une routine BASIC et un court programme en langage machine (à entrer par les programmes MACHDIM/POKEDIM du chapitre IV). Ils donnent exactement le même tracé sur papier, mais le sous-programme Basic (à appeler par un GOSUB 58000) est très lent.

```
58000 ' HARDCOPY (par TEST)
58010 ' AMSTRAD CPC - M.ARCHAMBAULT/86
58020 PRINT #8,CHR$(27);CHR$(64);
58030 PRINT #8,CHR$(27);"A";CHR$(7);
58040 PRINT #8,CHR$(27) "M";
58050 WIDTH 137
58060 FOR I=406 TO 14 STEP -14
58070 PRINT #8,CHR$(27)"L";CHR$(127);CHR
$(2);
58080 FOR J=0 TO 638
58090 T=TEST(J,I)*64+TEST(J,I-2)*32+TEST
(J,I-4)*16+TEST(J,I-6)*8+TEST(J,I-8)*4+T
EST(J,I-10)*2+TEST(J,I-12)
58100 PRINT #8,CHR$(T);:NEXT
58110 PRINT #8,CHR$(13):NEXT
58120 PRINT #8,CHR$(27);CHR$(64);
58130 RETURN
```

**LE PROGRAMME BINAIRE "HARD9F00"**

Son adresse départ est &9F00 ( = 40704) et sa longueur de 181 octets, donc il ne viendra pas empiéter sur tous ces petits programmes binaires que vous recopiez dans des revues, et qui eux commencent presque tous en &A000 ( = 40960). Ce 9F00 a été choisi parce que c'est facile à retenir, surtout quand on le rappelle dans le nom du programme... (une manière de faire qu'il est bon d'adopter).
Il vous faudra moins de dix minutes pour le saisir avec MACHDIM. Rappels : adresse départ &9F00 ou 40960, puis tapez dans l'ordre CD,A6,9F,3E, etc.
Pour le charger en RAM : (en début d'un programme)
MEMORY &9F00 – 1:LOAD"HARD9F00"
Pour exécuter un Hard Copy :
CALL &9F00.
Pour le sauvegarder :
SAVE"HARD9F00",B,&9F00,181

Même si c'est votre premier programme en langage machine, avec tous ces détails, vous n'avez pas le droit de vous rater !
Le listing que nous publions est en fait les deux colonnes de gauche d'une édition fournie par un programme assembleur. Rappelons encore que la colonne de gauche (adresses) ne sert dans MACHDIM qu'à vérifier que l'on ne s'est pas trompé.

**NOTE** : Du fait que l'interface imprimante des AMSTRAD ne ''sort'' que 7 bits au lieu de 8 (une économie à la fabrication), nos deux programmes déforment l'image en l'élargissant. C'est gênant pour les cercles qui deviennent des élipses. Ils fonctionnent indifféremment en MODE 0, 1 ou 2. Autre défaut : ils ne savent différencier les couleurs : c'est TEST = 0 ou TEST > 0. Avantages : faciles à adjoindre à un programme, peu encombrants en mémoire (181 octets pour HARD9F00...).

Si cela ne suffisait pas, il faudrait avoir recours à un logiciel du commerce, certes plus ''lourd'' à mettre en œuvre, mais nettement plus performant, comme par exemple ''TASCOPY'' de ''Sémaphore Logiciels'' qui ne présente pas les défauts ci-dessus, et qui convient aussi à la DMP 1.

Les illustrations graphiques du programme ''GRAPHMAT'' (chapitre VIII) ont été tracées par HARD9F00, tandis que les histogrammes du chapitre VI ont été obtenus par ''TASCOPY'' après redéfinition des couleurs.

```
9F00 CDA69F
9F03 3E1B
9F05 CD9D9F
9F08 3E31
9F0A CD9D9F
9F0D CDBABB
9F10 CDE7BB
9F13 32B49F
9F16 110000
9F19 218F01
9F1C 22B29F
9F1F 3E07
9F21 32B19F
9F24 3E0A
9F26 CD9D9F
9F29 3E0D
9F2B CD9D9F
9F2E 3E1B
```

```
9F30 CD9D9F
9F33 3E4C
9F35 CD9D9F
9F38 3E7F
9F3A CD9D9F
9F3D 3E02
9F3F CD9D9F
9F42 0E00
9F44 3AB19F
9F47 47
9F48 E5
9F49 C5
9F4A D5
9F4B CDF0BB
9F4E D1
9F4F C1
9F50 21B49F
9F53 BE
```

```
9F54 E1
9F55 37
9F56 2001
9F58 A7
9F59 CB11
9F5B 2B
9F5C 2B
9F5D 10E9
9F5F 3AB19F
9F62 FE07
9F64 2807
9F66 AF
9F67 CB11
9F69 CB11
9F6B CB11
9F6D 79
9F6E CD9D9F
9F71 13
9F72 E5
9F73 217F02
9F76 37
9F77 ED52
9F79 E1
9F7A 3805
9F7C 2AB29F
9F7F 18C1
9F81 23
9F82 7C
9F83 B5
9F84 2820
9F86 2B
9F87 110000
9F8A 22B29F
9F8D 3E07
9F8F BD
9F90 2092
9F92 7C
9F93 B4
9F94 208E
9F96 3E04
9F98 32B19F
9F9B 1887
9F9D CD2EBD
9FA0 38FB
9FA2 CD2BBD
9FA5 C9
9FA6 3E1B
9FA8 CD9D9F
9FAB 3E40
9FAD CD9D9F
9FB0 C9
9FB1 00
9FB2 00
9FB3 00
9FB4 00
```

Figure XVI 1
Programme HARD9F00.BIN

# Chapitre XVII

# LES PROGRAMMES D'ÉDITION

Il serait dommage d'utiliser votre belle imprimante uniquement pour des LIST # 8. Editer des tableaux bien présentés ou des étiquettes adhésives à la continue (mailing) est assez spectaculaire, mais autant vous prévenir tout de suite que la mise au point de ces programmes n'est pas chose facile ; du moins lorsque l'on veut sortir un travail impeccable. Vous en avez eu un avant-goût au chapitre XIV...

## CALIBRAGE DE L'IMPRIMANTE

Il est obligatoire de bien connaître son imprimante en ce qui concerne les diverses tailles de caractères. Le très court programme qui suit va nous fournir quatre réglettes graduées, qui vont s'avérer très utiles pour déterminer des tabulations horizontales, ainsi que les nombres exacts de caractères par ligne.
Enfin, mais cela ne vous servira normalement qu'une seule fois, vous pourrez régler très exactement le positionnement horizontal des tambours d'entraînement du papier.

```
10 ' CALIMP - CALIBRAGE D'IMPRIMANTE
20 PRINT #8,CHR$(27);CHR$(64)
30 WIDTH 137
40 A40$="1234567890123456789012345678901
234567890":' 40 CARACT
50 A80$=A40$+A40$:A160$=A80$+A80$
60 A160$=A80$+A80$
70 PRINT #8,"  120 CARACTERES EN PICA:"
80 PRINT #8,A80$;A40$:PRINT #8
90 PRINT #8,"  120 CARACTERES EN ELITE:"
```

```
100 PRINT #8,CHR$(27);"M";A80$;A40$:PRIN
T #8,CHR$(27);CHR$(64)
110 PRINT #8,"  160 CARACTERES EN CONDEN
SE:"
120 PRINT #8,CHR$(15);A160$:PRINT #8,CHR
$(27);CHR$(64)
130 PRINT #8,"  80 CARACTERES EN GROS:"
140 PRINT #8,CHR$(27);"W";CHR$(1);A80$
150 PRINT #8,CHR$(27);CHR$(64)
160 ' ------- FIN DE LISTING --------
```

On commence par vider le buffer (ligne 20) et fixer le WIDTH à 137. A noter que le WIDTH va demeurer actif dans l'AMSTRAD, non dans l'imprimante.
Normalement, vous devriez observer par ligne 80 caractères en PICA (= normaux), 96 en ELITE, 137 en condensé (attention ! parfois 132...) et 40 en GROS.
Vous remarquerez que l'interligne (le pas vertical) est toujours le même = 6 lignes par pouce, même en caractères larges.
Le pliage le plus courant est 11 pouces (un pouce = 25,4 mm) ; il y a donc 11 × 6 = 66 lignes par page. Ou 72 lignes sur le papier 12 pouces (= 304,8 mm), plus rare.
En fin d'édition, ligne 150, on revide le buffer de l'imprimante. Prenez toujours cette habitude au début et à la fin ; cela vous évitera bien des jurons...

## LA TABULATION HORIZONTALE

Sur imprimante, le LOCATE et le TAB ont leurs équivalences en codes EPSON, mais un conseil d'ami : ne vous en servez pas ! Car il suffit d'un rien pour que cela se dérègle, et c'est le désastre qui nourrira la corbeil à papier : les ESC ''e'', ESC ''f'' (ESC se programme par CHR$(27)), les CHR$(9) et CHR$(11), sont autant de pièges à .... loups à l'affût du moindre cas particulier pour vous jouer une belle farce !
La seule méthode vraiment sûre est de compter les caractères imprimés (par LEN) et les lignes. Exemple : pour imprimer A$ en TAB(15), on programme 14 blancs et A$ :

```
PRINT #8,SPC(14);A$
```

Examinez la ligne 6220 du programme d'édition du chapitre XIV : chaque chaîne est suivie d'un SPC qui la complète jusqu'à sa longueur maxi. N'ayez pas peur de programmer des calculs de SPC bien plus complexes ; dites-vous que le temps de calcul de l'AMSTRAD est infime vis-à-vis de celui que mettrait l'imprimante pour éditer une ligne sans calculs.

Pour centrer un titre TIT$ : soit NC le nombre de caractères par ligne. Il faut faire précéder TIT$ d'un nombre de blancs égal à la moitié de la différence NC-LEN(TIT$)

PRINT #8,SPC((NC – LEN(TIT$/2);TIT$

Même processus pour une marge à gauche, par exemple 10 caractères. On fixe alors :

M$ = SPACE$(10)

**NOTE :** SPACE$ peut définir une chaîne constituée par des blancs ; pas SPC:M$ = SPC(10) donnerait un "Syntax Error".

Pour la marge, chaque ligne commencera par :

PRINT #8,M$;... etc.

**Les caractères répétés**
Ils se programment par la fonction STRING$. Elle va bien souvent remplacer nos SPACE$ et SPC.
Voici un petit "problème" d'aspect délicat et pourtant si facile à résoudre.
Editer une table des matières où nous aurons 25 noms de longueurs inégales avec, à droite, le numéro de la page. Il faut que :
— les noms et les numéros de pages sont en colonnes ;
— chaque nom est "relié" à sa page par une série de points ;
— la longueur totale sera de 50 caractères ;
— l'édition devra être centrée.

SOLUTION :
80 – 50 = 30 donc marge M$ = SPACE$(15). Les noms sont en DIM N$(25), les pages en DIM P(25).
On va prévoir trois "cases" pour les numéros de page, alors que deux suffisent, afin que les points ne "touchent" pas les chiffres. Il reste donc 50 – 3 = 47 cases pour nom + points. D'autre part, on va aussi laisser un blanc entre le nom et les points.
La longueur de la série de points sera donc égale à 46 – longueur du nom.
Tout cela se programme alors :

```
410 FOR I=1 TO 25
420 PRINT#8,M$;N$(I);" ";STRING$(46-LEN(N$(I)),".");
430 PRINT #8,USING "###";P(I)
440 NEXT
```

## LA TABULATION VERTICALE

Il est impossible de régler le papier de telle sorte que l'impression

démarre exactement sous la pliure du papier (afin de compter 1 à 66). Le repère fixe dont vous disposez est situé à combien de lignes au-dessus de la tête d'impression ? Pour le savoir, faites une petite réglette verticale :

```
FOR I=0 TO 20:PRINT#8,I:NEXT
```

Ensuite, descendez le papier jusqu'à ce que le ''0'' imprimé soit en face de votre repère fixe. Puis tapez :

```
PRINT #8,''   DEPART''
```

Vous lirez alors ce fameux nombre de lignes d'amorçage feuille. Supposons cinq.

Revenons à notre table des matières : nous voulons en plus un titre ''INDEX'' en gros caractères, trois lignes au-dessus et nous voulons que cette page soit aussi centrée verticalement.

SOLUTION
1 ligne titre + 2 lignes blanches + 25 lignes de texte = 28 lignes imprimées. La marge en haut est donc de :

$$(66-28)/2=19$$

et autant pour le marge en bas.
Puisque l'on démarre avec 5 lignes déjà avancées, il faudra en sauter 14. En revanche, à la fin du texte, il faudra sauter 19 + 5 = 24 pour que la nouvelle pliure se positionne exactement en face de notre repère. On pourrait alors reprendre une autre édition sas repositionner le papier à la main (saut de page automatique).

```
390 FOR I=1 TO 14:PRINT#8:NEXT
400 PRINT#8,CHR$(14);SPC(17);''INDEX'':PRINT#8
410
...  (déjà écrites)
440
450 FOR I=1 TO 24:PRINT#8:NEXT
```

Au total, cela nous fait SEPT lignes de Basic pour éditer proprement cette table des matières, et avec saut de page auto.
Etait-ce long ? Etait-ce super-compliqué ?

**NOTE** : Le CHR$(14) fait imprimer en gros caractères sur *une ligne seulement.*

## ENCADREMENTS DE TABLEAUX

Par encadrement, nous entendons le tracé de traits horizontaux et verticaux. Ce sont ces derniers qui paraissent les moins continus en rai-

son de l'interligne de l'imprimante. Dans le tableau d'édition de ''LOGI-CLAS'' (chapitre XIV), le signe ''barre verticale'' nous a fourni des traits verticaux en pointillés. C'est acceptable mais on peut faire mieux.
La méthode consiste à établir une ligne dite de séparation faite de barres espacées de blancs pour la tabulation. Cette ligne est imprimée en *demi-interligne* afin que les traits verticaux se chevauchent en un trait continu. Autrement dit, on commande à l'imprimante 12 lignes par pouce au lieu de six, puis on alterne une ligne ''normale'', une ligne de ''séparation'' : les lignes de caractères sont donc toujours à l'écartement normal.
Les traits horizontaux restent à base de tirets (signe moins), mais alors se pose le problème des extrémités gauche et droite : si on y met un tiret, il y aura un léger dépassement de part et d'autre du trait vertical. Si on y met une barre verticale, il y aura dépassement en haut et en bas pour les traits horizontaux délimitant le haut et le bas du tableau. Le compromis, c'est le '':''.
Le programme de démonstration qui suit, ''TABLETIQ'' comprend trois parties :
- un ''remplissage'' (4 fiches...) d'un tableau DIM P$ (lignes 20 à 1407 que nous éditerons ;
- l'édition en tableau avec encadrement ;
- ces mêmes données (noms, adresses bidon) sont imprimées sur des étiquettes auto-collantes (mailing).

**Le programme tableau**
**(lignes 200 à 380)**

Il commence bien sûr par définir les chaînes horizontales : le séparateur SEP$ et le trait continu TC$. Vous remarquerez qu'ils sont paramétrés en fonction du nombre et de la longueur des rubriques ; ce sont donc des lignes que vous pourrez réutiliser pour vos programmes.
En ligne 300 commence l'édition : trait horizontal, légendes des rubriques (indice zéro du DIM P$), trait horizontal, impression du contenu de chaque fiche (GOSUB 2000) et le trait horizontal final.

```
10 ' TABLETIQ - TABLEAUX & ETIQUETTES
-   M.A / 86
20 ' REMPLISSAGE DIM
50 DIM P$(200,4),LR(4):NF=4
60 DATA NOM,PRENOM,ADRESSE,CODPO,VILLE
70 DATA MR VERGNE,PAUL,58 RUE TUPIN,2635
0,CREPOL
80 DATA MR COISE,SIMON,12 PLACE D'ARMES,
38420,DOMENE
90 DATA MLLE GUILLON,SOPHIE,136 BD DUQUE
SNE,47320,CLAIRAC
```

```
100 DATA MME PRASEY,MARTINE,56 AVENUE FO
CH,53000,LAVAL
110 DATA 15,10,20,5,12
120 FOR N=0 TO NF:FOR R=0 TO 4
130 READ P$(N,R):NEXT:NEXT
140 FOR I=0 TO 4:READ LR(I):NEXT :'LARGE
URS RUBRIQUES
200 ' TRAITS HORIZONTAUX
210 FOR I=0 TO 3:SEP$=SEP$+SPACE$(LR(I))
+" ! ":NEXT:SEP$="! "+SEP$+SPACE$(LR(4))
+" !" : ' SEPARATEUR LIGNE
220 LT=LEN(SEP$) :' LARGEUR TOTALE
230 TC$=":"+STRING$(LT-2,"-")+":" :'TRAI
T CONTINU
300 ' EDITION TABLEAU
310 PRINT #8,CHR$(27);CHR$(64)
320 PRINT #8,CHR$(27);"A";CHR$(6);:' DEM
I-INTERLIGNE
330 PRINT #8,TC$:PRINT #8,SEP$
340 N=0:GOSUB 2000:' LEGENDES
350 PRINT #8,SEP$:PRINT #8,TC$:PRINT #8,
SEP$
360 FOR N=1 TO NF:GOSUB 2000:PRINT #8,SE
P$:NEXT
370 PRINT #8,TC$
380 PRINT #8,CHR$(27);CHR$(64):FOR I=1 T
O 10:PRINT #8:NEXT
400 ' ETIQUETTES MAILING
410 NC=80 :' Nombre Caract. par Ligne
420 NE=3  :' Nombre Etiquettes par Rang
430 PV=9  :' Pas vertical en Lignes
440 LE=NC/NE :' Largeur Etiquette en Car
acteres
450 WIDTH NC:FOR N=1 TO NF STEP NE
460 ' LIGNE NOM+PRENOM
470 FOR F=N TO N+NE-2
480 A$=P$(F,0)+" "+P$(F,1):GOSUB 3000:NE
XT
490 A$=P$(F,0)+" "+P$(F,1):GOSUB 3500
500 ' LIGNE ADRESSE
510 FOR F=N TO N+NE-2
520 A$=P$(F,2):GOSUB 3000:NEXT
530 A$=P$(F,2):GOSUB 3500
```

```
540 PRINT #8 :' LIGNE DE SEPARATION
550 ' LIGNE CODE POSTAL+VILLE
560 FOR F=N TO N+NE-2
570 A$=P$(F,3)+"  "+P$(F,4):GOSUB 3000:NE
XT
580 A$=P$(F,3)+"  "+P$(F,4):GOSUB 3500
590 ' SAUT VERTICAL D'ETIQUETTES
600 FOR I=1 TO PV-4:PRINT #8:NEXT
610 NEXT N
620 END
2000 ' LIGNE DU TABLEAU
2010 PRINT #8,"! ";:FOR R=0 TO 4
2020 PRINT #8,P$(N,R);SPC(LR(R)-LEN(P$(N
,R)));" ! ";:NEXT
2030 RETURN
3000 'LIGNE ETIQUETTES GAUCHE+CENTRE
3010 PRINT #8,SPC((LE-LEN(A$))/2);A$;SPC
((LE-LEN(A$))/2);
3020 RETURN
3500 'LIGNE ETIQUETTE DROITE
3510 PRINT #8,SPC((LE-LEN(A$))/2);A$
3520 RETURN
65535 ' ------- FIN DE LISTING ------
```

## L'IMPRESSION D'ETIQUETTES A LA CONTINUE

L'imprimante est maintenant chargée avec des étiquettes en bandes. Deux utilisations principales : les étiquettes-adresses ( = ''mailing'') et étiquetage banal de flacons, etc.
Dans la largeur de papier pour machines amateurs (24 cm), on trouve plus facilement du ''Deux étiquettes'' par rangée (107 × 35 mm) que du ''Trois étiquettes''. Qu'importe ! Notre programme est paramétrable pour un nombre quelconque d'étiquettes par rangée, quelle que soit la largeur du papier ou la taille des caractères !
On vous demande simplement d'entrer en lignes 410 à 430 le nombre de caractères par ligne, le nombre d'étiquettes par rangée et le pas vertical entre deux étiquettes exprimé en *lignes.* Si vous ne savez pas, mesurez-le en millimètres et divisez par 4,23 (1/6 pouce ≃ 4,23 mm).
Le problème de ce type d'édition est assez spécial. Il faut en effet imprimer sur une même ligne la même rubrique de deux ou trois fiches successives, afin qu'une étiquette reçoive le contenu d'une fiche mais en plusieurs lignes.

Ajoutez à cela que nous exigeons un *centrage automatique* de chaque rubrique sur une étiquette, que deux rubriques peuvent être sur la même ligne, exemple code postal et ville, que la dernière fiche peut entamer une rangée sans obligation de la terminer, et vous avez une idée de la complexité de nos boucles FOR NEXT...
Et pourtant, ce programme, facilement transformable, est assez court (et efficace...).
Nous avons mis de nombreux REM afin que vous compreniez bien la méthode : on prépare la chaîne A$ à imprimer avec ses blancs, pour son centrage, devant et derrière. A la suite, la même rubrique mais de la fiche suivante. S'il s'agit de la dernière étiquette, c'est uniquement chaîne d'espaces + A$, pour le retour ligne.. Rubrique suivante, etc.
Lorsqu'une rangée d'étiquettes est terminée, le papier avance pour présenter la rangée suivante.

Quelques détails expliqués :
- Chaque ligne-rubrique est traitée à part et successivement ; ainsi nous avons ligne 1 = nom, espace, prénom ; ligne 2 = adresse ; ligne 3 = rien ; ligne 4 = code postal, espace, ville.
- Lignes 490, 530 et 580, il semble que l'on rappelle la même fiche F : il n'en est rien car auparavant nous avions un FOR F = N TO ... : NEXT. Rappelez-vous qu'après un NEXT la variable est égale à la valeur limite + 1 (MPSA chapitre XI).
- Saut d'étiquettes : elles sont au pas de PV lignes, nous avons imprimé quatre lignes (ligne 600) ; il nous faut donc avancer de PV – 4 lignes pour commencer une nouvelle rangée d'étiquettes à la même hauteur que la première.

A présent, modifiez la ligne 420 en tapant NE = 2 étiquettes par rangée et repartez par GOTO 400. Vous constaterez que la disposition deux par rangée est tout aussi parfaite.
Pour vous assurer du côté universel de ce programme, vous allez simuler du quatre étiquettes par rang en écriture condensée !

```
405 PRINT #8,CHR$(15);
410 NC=137
420 NE=4
```

et GOTO 400. Vous vérifiez alors que tous les centrages automatiques sont respectés.
Et en caractères ELITE ? Ça marche aussi :

```
405 PRINT#8,CHR$(27);"M";
410 NC=96
```

Et idem en gros caractères :

```
405 PRINT#8,CHR$(27);"W";CHR$(1);
410 NC=40
420 NE=2
```

**NOTE** : Dans notre listing nous n'utilisons que les caractères PICA (normaux). Pour des éditions en autres caractères, n'oubliez surtout pas en final de vider le buffer de l'imprimante.

```
615 PRINT # 8,CHR$(27);CHR$(64)
```

Sur des étiquettes situées sur la droite, vous pouvez parfois observer des centrages incorrects. C'est dû aux calculs des SPC qui sont traduits en nombres entiers, d'où des décalages qui peuvent s'ajouter (ou se retrancher) sur une même ligne. Ce sera toujours mieux qu'avec certains logiciels ''professionnels'' pour IBM PC où toutes les rubriques partent du bord gauche de l'étiquette...

| NOM | PRENOM | ADRESSE | CODPO | VILLE |
|---|---|---|---|---|
| MR VERGNE | PAUL | 58 RUE TUPIN | 26350 | CREPOL |
| MR COISE | SIMON | 12 PLACE D'ARMES | 38420 | DOMENE |
| MLLE GUILLON | SOPHIE | 136 BD DUQUESNE | 47320 | CLAIRAC |
| MME PRASEY | MARTINE | 56 AVENUE FOCH | 53000 | LAVAL |

```
MR VERGNE PAUL          MR COISE SIMON          MLLE GUILLON SOPHIE
 58 RUE TUPIN          12 PLACE D'ARMES           136 BD DUQUESNE

 26350 CREPOL           38420 DOMENE               47320 CLAIRAC


MME PRASEY MARTINE
 56 AVENUE FOCH

  53000 LAVAL
```

# Chapitre XVIII

# ÉCONOMIES ET PANNES DE MÉMOIRE

Nous avons préféré attendre la fin de cet ouvrage pour parler de ces choses ''graves''. En effet, il est normal que l'utilisation d'un programme utilitaire conduise souvent à une importante occupation mémoire. On se dit ''mon programme fait 10 kilo-octets, mon fichier 12 Ko ; total 22, je suis tranquille...''. Erreur ! tout se passe bien au début de la séance d'exploitation et puis, sans prévenir, *plus de curseur* le clavier ne répond plus ! même pas ESC ! Vous venez de saturer la mémoire avec des variables provisoires de travail. Le micro fait lui-même son FRE('' ''), et cela peut prendre de nombreuses minutes avant qu'il ne vous ''rende la main'', et son curseur.
N'accusez pas AMSTRAD : ce phénomène est commun à *tous* les micro-ordinateurs.
Il n'y a malheureusement pas de trucs pour se mettre à l'abri de ce plantage (temporaire) de mémoire. En revanche, il y a de multiples précautions pour que cela n'arrive que le plus tard possible...
Et ne croyez surtout pas que l'achat ''d'extensions mémoire'' de 64, 128, 256 ou 512 kilo-octets va vous tirer d'affaire ! Cela n'a rien à voir : le microprocesseur des AMSTRAD est un ''8 bits'', donc il ne peut ADRESSER que de 1 à 65535 octets (= 64 kilo-octets : 1 kilo-octet = 1024 octets et non 1000). Les extensions sont des ''disques RAM''.
Avant de se lancer dans les conseils, il est préférable de décrire sommairement l'organisation de la mémoire des CPC, car c'est assez original.

## LA REPARTITION DE LA MEMOIRE

Les adresses vont de 1 à 65535 (&FFFF en hexa). Parlons en décimal, ici c'est plus clair. Le haut de la mémoire, à partir de 49152, c'est l'écran vidéo. Ça ne bouge pas.

En bas de la mémoire, la place disponible commence à 368 ; c'est à partir de 368 que va se charger un programme Basic. C'est une adresse fixe elle aussi.
Le reste est plutôt ''mouvant''.
Revenons vers le haut. Avec un 464 sans DOS et en tapant SYMBOL AFTER 255, puis en demandant PRINT HIMEM, on a 44023. C'est l'Himalaya des adresses disponibles.
Un CPC 464 avec DOS, à la mise sous tension, on a HIMEM 42619. Si on rogne un peu avec un SYMBOL AFTER 255, on atteint 42739, soit 120 octets de plus. Faisons alors PRINT FRE(0) afin de connaître la mémoire disponible = 42369. La différence de 370, c'est la zone super basse, sous le Basic.
A chaque fois que l'on définit une variable, par exemple A$ = ''MACHIN'', ça va aller se ''coller au plafond'' sous le HIMEM. A présent disons A$ = ''TRUC'', il n'empêche que les octets occupés par les caractères du mot ''MACHIN'' n'ont pas bougé ; on a mis ''TRUC'' collé après. Donc la zone mémoire des variables *gonfle en descendant.*
Dans cette zone de variables (en haut, rappelons-le), il y a des variables ''en action'', d'actualité, et des valeurs ''mortes'' (périmées).
La zone du bas occupée par le programme Basic *ne varie pas,* mais il va arriver un moment où la zone des variables qui grossit va l'atteindre et c'est le blocage. Alors c'est le processus du FRE('' '') qui va ''vider le cimetière'' des valeurs mortes en ne laissant que les valeurs en service. La zone des variables va se rétrécir vers le haut ; oui mais ça peut demander plus de cinq minutes !

## LE CAS DE L'OPENOUT OU OPENIN

Pour exécuter l'une de ces commandes, l'AMSTRAD a besoin de 4 kilo-octets. Alors il va descendre le HIMEM d'autant, et tout le paquet de variables qu'il y a au dessous, et s'il est déjà gros, cela peut prendre des minutes. Après le CLOSEOUT ou CLOSEIN, le HIMEM va reprendre son ancienne valeur, en remontant la zone des variables ; encore des minutes.
Afin d'éviter cette double farce, on abaisse tout de suite le HIMEM environ 4000 octets plus bas par la fameuse ligne :

OPENOUT ''BIDON'':MEMORY HIMEM – 1:CLOSEOUT

Ainsi MEMORY a fixé définitivement le HIMEM a sa valeur mini qu'il prendrait. Là, c'est ultra rapide puisqu'il n'y a pas encore de variables à ''translater''.
Faisons alors PRINT HIMEM:38642, soit 38272 octets disponibles. Quelle perte ! Mais on ne peut rien y faire.
Conclusion : le fait d'avoir à utiliser OPENIN et OPENOUT pour charger et re-enregistrer un ou des fichiers coûte 4 kilo-octets de réservation mémoire.
**NOTE** : Si vous avez à définir des SYMBOL, faites-le AVANT l'OPENOUT ''BIDON'', sinon plantage immédiat.

## LE CAS DES PROGRAMMES BINAIRES

Un programme binaire ne peut se charger dans une zone déjà occupée, sinon message ''Memory Full''.
Ainsi avant de charger le programme binaire ''TOTO'' en &A000, il est ultra prudent (souvent obligatoire) d'écrire :

MEMORY &A000 – 1:LOAD''TOTO''

Les CPC 464, 664 et 6128 ont des ''plans mémoire'' légèrement différents. La grande différence est, nous l'avons vu, lorsque l'on équipe un 464 de l'interface lecteur de disquettes ; le DOS se prend 2 kilo-octets.
Voilà qui vous explique pourquoi des logiciels en langage machine sur cassettes ne peuvent se transférer sur disquette. Pourquoi tel programme sur disquette fonctionne sur 464 et pas sur 6128 ?
Le programmeur en langage machine a trop tendance à loger ses programmes le plus haut possible. La dernière adresse de son programme le plus haut n'est plus bien loin de la ''frontière''. Or, si le fabricant AMSTRAD sort un nouveau modèle avec des ''trucs'' en plus qui abaissent de ce fait le plafond...
Vous devinez la suite. Si vous avez conservé un enregistrement du ''programme source'', il vous sera facile de seulement modifier le ORG, et de ré-assembler ; sinon il faudra désassembler le ''programme objet'' et retaper le ''source''.
Bon courage ! Et tout cela parfois pour une trentaine d'octets trop haut...
Un exemple de précaution pour routines courtes. Le programme binaire de Hard-copie du chapitre XVI a été *tout de suite* assemblé et sauvegardé en deux versions d'adresse départ : &9F00 et &A444.

## EVITONS LES GASPILLAGES PAR RESERVATIONS INUTILES

Rappelons une fois de plus qu'il faut définir le maximum de variables numériques comme étant des nombres entiers. Par exemple DEFINT I – N ou DEFINT A – E,I – N ou totalement DEFINT A – Z. Deux avantages à cela : 5 octets au lieu de 2 par nombre défini. Des boucles FOR NEXT (et tris) deux fois plus rapides !
Les inconvénients des entiers sont de deux sortes : pas de décimales et limités entre ± 32767. Plusieurs parades à cela :

- Un prix PR exige 2 décimales. On le multiplie par cent pour le stocker en entier. Rien n'empêche alors d'afficher ou d'imprimer PR/100, ou en final de définir PR! = PR/100.
- On applique la technique inverse (diviser par 10, 100, 1000...) pour des valeurs supérieures à 32000.
- Une variable est toujours positive, mais peut aller de 0 à 65000 : on ''récupère'' la zone négative des entiers N% = N – 37000 ; que l'on restituera en final, mais tous les N% auront été mémorisés sur deux octets.

Un autre gaspillage classique concerne les tableaux DIM surdimensionnés : trop de lignes prévues. On n'a pas utilisé la colonne et la ligne zéro. Exemple : le simple fait de déclarer (sans remplir) DIM A$(300,8) coûte... 8148 octets ! Et maintenant DIM N(300,8) avec des nombres réels : 13566 octets. Avec des nombres entiers DIM N%(300,8), la réservation est moins gourmande, 5439 octets.
C'était prévisible : 301 × 9 = 2709 ''cases''. Multiplié par deux octets = 5418, et par cinq octets : 13545. Le DIM A$(300,8) a réservé trois octets (soit trois caractères) par case. Lui va grossir en occupation mémoire au fur et à mesure de son remplissage, alors que les DIM numériques resteront fixes.
Dans un ''grand'' DIM numérique, la gaffe est de définir une colonne comme étant le résultat d'un calcul simple entre deux autres. Exemple : prix HT et % TVA suffisent, francs TVA et prix TTC sont deux colonnes de DIM dont on peut se dispenser. On les calculera au fur et à mesure des besoins. Sur papier, des colonnes étaient indispensables, pas en informatique.
Evitez de même les variables intermédiaires dans un calcul, un traitement de chaîne. Certes le listing est plus clair, l'écriture Basic est plus simple car elle dispense de rabâcher des longs tronçons pleins de parenthèses ; mais si jamais cela se trouve au cœur de plusieurs boucles FOR NEXT imbriquées, pensez au cimetière des variables...

**La concaténation**
C'est un gouffre d'octets. Exemple : nous voulons créer une chaîne C$ qui sera la suite des 58 caractères entre A et z.

```
FOR I=65 TO 122:C$=C$+CHR$(I):NEXT
```

Cett ligne banale a consommé, gaspillé 1737 octets ! Pourquoi ? Parce qu'au cimetière, nous avons empilé pour C$ A, AB, ABC, ABCDE, etc., d'où ce gâchis que nous a montré un PRINT FRE(0) avant et un autre après cette ligne.

## L'EXPLOITATION PRUDENTE

Prenons le cas d'un gestionnaire de fichiers avec en RAM un fichier de 10 à 20 kilo-octets. Tant que l'on fait de la saisie ou de la recherche, le risque du blocage mémoire est raisonnable. Si cela arrivait (plus de curseur), ne touchez surtout pas au clavier, tout est néanmoins mémorisé dans le buffer clavier et serait exécuté en fin de ''vidange''.
En revanche, le risque est énorme avec les phases d'enregistrement ou sauvegarde et les tris. En effet, ces quelques lignes Basic ne sont que des FOR NEXT imbriqués, et au milieu plusieurs noms de variables qui vont changer de valeur des milliers de fois...
Comment ''faire le ménage'' dans le cimetière ? Deux méthodes :
— Des PRINT FRE('' '') ou XXX=FRE('' '') assez fréquents, c'est-à-dire sans attendre l'envahissement, car l'attente serait très longue. L'usage d'un EVERY 100 GOSUB 4000 et en 4000 XXX=FRE(''

''):RETURN serait désastreux dans notre progamme LOGICLAS, car dès le premier chargement de fichier vous resteriez de longues minutes à attendre au lieu de quelques secondes.

- Le RUN remet, lui, immédiatement la zone des variables à zéro. Oui, mais on perd tout !

De deux maux choisissons le moindre :
Imaginez qu'après avoir chargé votre fichier, vous vouliez en faire trois versions triées différemment et les enregistrer. Si votre fichier ne fait que 5000 octets, pas de risques. S'il fait plus de 100 fiches, vous n'avez aucune chance de finir votre travail sans le fameux blocage de dix minutes.
Voici la méthode barbare mais rapide et sûre (pour lecteur de disquettes) :
a — charger le fichier ''brut'',
b — faire le tri n° 1,
c — l'enregistrer sur disquette,
d — quitter et taper RUN,
e — charger le fichier brut,
f — faire le tri n° 2,
etc.

Je dois vous surprendre, mais votre travail sera fait en moins de cinq minutes au lieu de près d'un quart d'heure...

## CONCLUSION

Du fait que l'AMSTRAD est doté d'un microprocesseur 8 bits, le célèbre Z80, il ne peut pas travailler correctement sur des fichiers qui approchent les 20 kilo-octets. La machine n'est plus appropriée, autant essayer de faire un déménagement avec une voiture Break... S'il vous est impossible de scinder votre énorme fichier en plusieurs tailles raisonnables, la seule solution est de revendre votre 8 bits pour un 16 bits qui, cas des IBM PC et de ses ''compatibles'', n'ira pas plus vite mais peut adresser jusqu'à 640 kilo-octets.... Autrement dit, l'AMSTRAD a tous les outils pour faire du bon professionnel, mais la taille de sa mémoire le limite à l'artisanat, en lui interdisant les ''grands chantiers''.
Il fallait que cette évidence soit dite.

*Photocomposition : FIDELTEX*
*Maquette : SORACOM*
*Impression : L'INDEPENDANT*
*Numéro d'éditeur : 054*


**ISBN 2 904032-43-6**

**Votre AMSTRAD sait aussi faire autre chose que de ''lasériser'' des extra-terrestres ; il sait se rendre utile pour résoudre vos petits problèmes, ou écourter de façon spectactulaire des besognes courantes et fastidieuses. Reste la question du logiciel : ceux du commerce sont souvent très difficiles à adapter à votre besoin précis, voire inexistants. Il faut le programmer soi-même.**

**Ce livre veut être un utilitaire pour faire des utilitaires ; il contient davantage de méthodes, d'astuces, de mises en garde, que de logiciels genre ''PRET-A-LOADER''. En revanche, beaucoup de routines courtes et passe-partout d'usage fréquent et des programmes plus longs mais conçus pour être modifiés en fonction des besoins.**
**A l'exception d'un seul, tous nos programmes sont en Basic, donc à la portée de tous.**



**PRIX : 85 Frs**


PROGRAMMES UTILITAIRES POUR AMSTRAD
Michel Archambault



Document numérisé avec amour par

AMSTRAD CPC

MÉMOIRE ÉCRITE



https://acpc.me/